NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IMPROVED GEO-SPATIAL RESOLUTION USING A MODIFIED APPROACH TO THE COMPLEX AMBIGUITY FUNCTION (CAF)

by

Glenn D. Hartwell

September 2005

Thesis Advisor: Herschel H. Loomis, Jr.
Second Reader: Alan A. Ross

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Improved Geo-Spatial Resolution Using a Modified Approach to the Complex Ambiguity Function (CAF). Upper and Lower Case

5. FUNDING NUMBERS

6. AUTHOR(S) Glenn D. Hartwell

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

The purpose of this thesis is to implement the CAF-Map method of geolocation in MATLAB. This method is a modification to the traditional Cross Ambiguity Function (CAF) based TDOA, FDOA geolocation where TDOA and FDOA are determined by locating the peak in the CAF plane and then the peak's information is fed to a Least Squares like geolocation tool to determine the emitters geolocation. This method omits the step in which the geolocation is determined with the "post processed" CAF peak information and instead maps the CAF surface directly to the earth's surface. In this thesis, the traditional CAF based geolocation is explained and the limitations are discussed. After this, the development of the CAF-Map method is explained and the algorithm is presented. This thesis explores the use of the CAF-Map method as a geolocation alternative to the traditional TDOA, FDOA methods and demonstrates its ability to geolocate co-channel emitters.

14. SUBJECT TERMS
Cross Ambiguity Function, CAF, Geolocation, Time Difference of Arrival, TDOA, Frequency Difference of Arrival, FDOA

15. NUMBER OF PAGES
119

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

IMPROVED GEO-SPATIAL RESOLUTION USING A MODIFIED APPROACH TO THE COMPLEX AMBIGUITY FUNCTION (CAF)

Glenn D. Hartwell
Civilian, Naval Research Laboratory, Washington, D.C.
B.S., Pennsylvania State University, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2005

Author: Glenn D. Hartwell

Approved by: Herschel H. Loomis, Jr.
Thesis Advisor

Alan Ross
Second Reader

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The purpose of this thesis is to implement the CAF-Map method of geolocation in MATLAB®. This method is a modification to the traditional Cross Ambiguity Function (CAF) based TDOA, FDOA geolocation where TDOA and FDOA are determined by locating the peak in the CAF plane and then the peak's information is fed to a Least Squares like geolocation tool to determine the emitters geolocation. This method omits the step in which the geolocation is determined with the "post processed" CAF peak information and instead maps the CAF surface directly to the earths surface.

In this thesis, the traditional CAF based geolocation is explained and the limitations are discussed. After this, the development of the CAF-Map method is explained and the algorithm is presented. This thesis explores the use of the CAF-Map method as a geolocation alternative to the traditional TDOA, FDOA methods and demonstrates its ability to geolocate co-channel emitters.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

There are several people I would like to thank that have helped me throughout my work on this thesis. First, I would like to thank my advisor, Professor Herschel Loomis, my second reader, Professor Alan Ross, and my Naval Research Laboratory advisor, Dr. Kenneth Clark, for their patience, guidance, and support. I would also like to thank Mr. Al Buczek, Mr. Dave Pettit, and Mr. Brad Kuhn of the Naval Research Laboratory for their guidance and expertise on the topic of emitter geolocation as well as their support during the time spent on this thesis. Lastly, I would like to thank my wife for her support and keeping me sane.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The task of geolocating radio frequency emitters has many uses. One of the most popular methods of geolocation is a combined Time Difference of Arrival (TDOA) and Frequency Difference of Arrival (FDOA) method utilizing the Cross Ambiguity Function (CAF) to generate the TDOA and FDOA values. This method does not have a direct solution and requires that a numerical estimation method be employed to determine the emitter's geolocation. The method studied in this thesis eliminates the numerical estimation required by the traditional method and instead uses an alternative method that maps the TDOA and FDOA values generated by the CAF function directly to an X, Y coordinate value. This method is called the CAF-Map method and was first explored by Mr. Al Buczek of the Naval Research Laboratory in Washington DC [1].

The CAF-Map technique relies on the fundamental principle that primary correlation peaks for stationary emitters will be perfectly consistent for all CAF surfaces. In effect, all available CAF surfaces are mapped and combined in a common geographic frame which results in an image analogous to radio imaging. The apparent position of spurious artifacts, secondary side lobes, and the left-right images will lack the consistency of the true peaks due to the varied geometry and dynamics of the collection platforms.

In this method, the entire geographic coverage area's TDOA(s) and FDOA(s) are computed to form a lookup table for each snapshot. Then, each snapshot's CAF is computed over the range of the expected TDOA(s) and FDOA(s). Once the CAF(s) are computed for each snapshot, a geographic MAP can be formed using the lookup tables to "map" the CAF to the ground. Summing each "map" over a common geographic area yields a RF energy map of the area for the collected frequency. This method produces a geographic "image" of the geolocated energy rather than a traditional map showing an error ellipse for the emitter's estimated location. Figure 1 shows the resulting image using the CAF-Map method of geolocation.

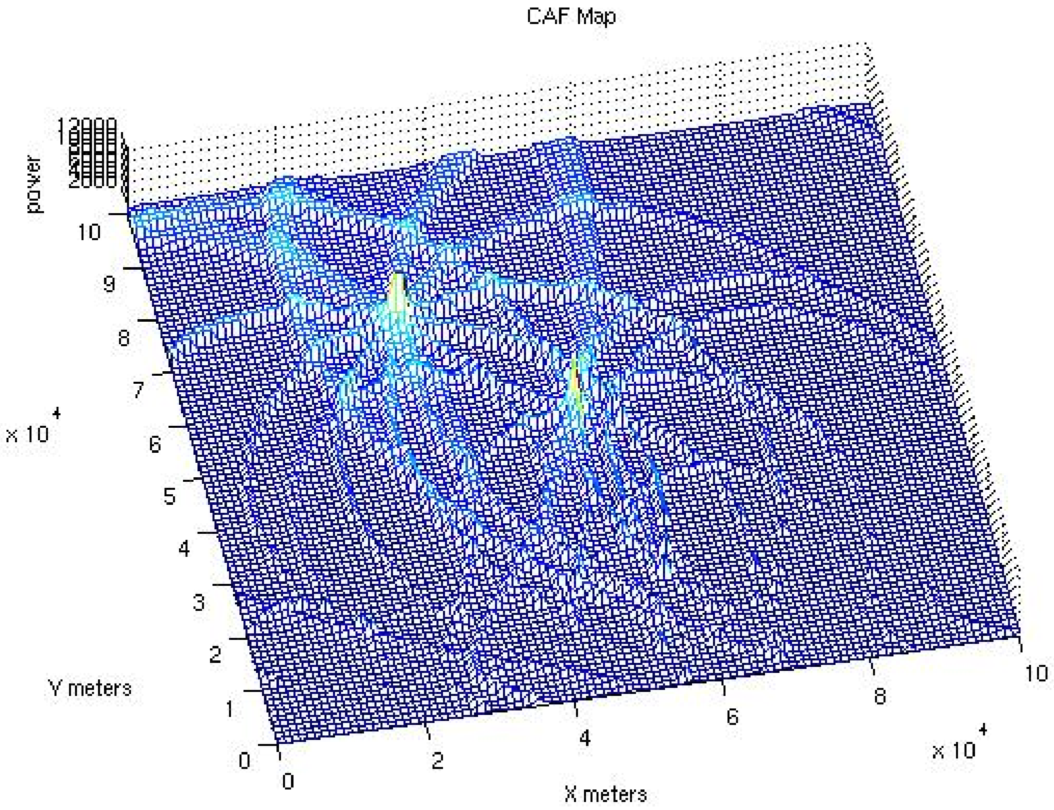


Figure 1: Example of the CAF-Map geolocation of two emitters

I. INTRODUCTION

A. BACKGROUND

The topic of emitter geolocation is critical for both military and commercial applications. To date, this work has been limited primarily on single emitters that are isolated or those that are dominant in the processing bandwidth. Most military systems have focused on pulsed emitter geolocation using Time Difference Of Arrival (TDOA) techniques to determine the emitter's location. However, as RADAR systems become more advanced, the use of Continuous Wave (CW) waveforms is increasingly more prevalent. This means that the traditional TDOA methods that rely on determining the pulses Time Of Arrival (TOA) are failing to satisfy user requirements. To geolocate both modern CW RADAR systems as well as communications signals, different geolocation methods are required. Most commonly, the geolocation of these CW emitters is performed using the combination of Time Difference Of Arrival (TDOA) and Frequency Difference Of Arrival (FDOA) measurements derived from the Cross Ambiguity Function (CAF). The CAF requires the processing of simultaneous Pre-Detection (Pre-D) collected signals of a single emitter from spatially separated collection positions to determine the TDOA and FDOA. These TDOA and FDOA measurements are used to determine the emitter's geolocation.

Difficulties arise in standard TDOA/FDOA geolocation processing when faced with the problem of separating emitters that are co-channel (occupying the same frequency) and geographically close to one another. This problem can be best addressed as a problem of geo-spatial resolution rather than as a geolocation accuracy problem.

B. OBJECTIVE

The main topic of this thesis is to study a technique put forth by Mr. Al Buczek of the Naval Research Laboratory during the early 1990's that could improve the co-channel geo-spatial resolution of the CAF process. This technique was never fully explored at the time due to the computational power required and other higher priority research topics. This method is called the CAF-MAP [1] where the CAF surface is mapped directly to the Earth's surface. This thesis will study this technique and produce MATLAB®

simulations that validate the technique as an alternative to the traditional TDOA/FDOA geolocation methods. It is not the objective of this thesis to compare the results for these two techniques but only to illustrate the functionality of the CAF-Map method.

C. RELATED WORK

There exist numerous papers, thesis, and dissertations on the topic of radio frequency emitter geolocation. Most of works in common with this thesis are on the topic of the CAF technique. Stein's paper [2] is considered to be the paramount paper on the subject of computing the CAF surface. Other papers [4], [5], and [6] relate to the traditional method of TDOA and FDOA geolocation techniques.

D. THESIS ORGANIZATION

This thesis is organized into six chapters. Chapter II provides a short description of the Cross Ambiguity Function (CAF) with discussions on the performance of the TDOA and FDOA measurements and the error behavior of the measurements. Chapter II also discusses the effects of the collectors' geometry on the TDOA and FDOA generated by the CAF process. Chapter III discusses the traditional TDOA and FDOA geolocation method including the Newton-Raphson method of estimation, the derivation of the Weighted Least Squares method, and a discussion of the 95% confidence ellipse. The CAF-Map method is described in Chapter IV. This chapter includes a description of the MATLAB® CAF-Map function and the generation of the TDOA and FDOA lookup tables. Chapter IV also includes discussions on the computation of the CAF surface, the mapping of the CAF surface to the X, Y coordinate system, and a description of the resulting surface. This chapter includes the generation of the test signals as well. Chapter V shows several examples of the CAF-Map method geolocating simulated signals using different collection geometries and number of co-channel emitters. And finally Chapter VI summarizes the results of the thesis and discusses possible follow on work based on this thesis.

II. THE CROSS AMBIGUITY FUNCTION

A. A SHORT EXPLANATION OF THE CAF

The Cross Ambiguity Function (CAF), sometimes known as the Complex Ambiguity Function (CAF), is simply the correlation of two signal waveforms over a range of time and frequency offsets. The most common expression for the CAF is shown in Equation (2-1) and was derived by Stein [2].

$$A(\tau, f) = \int_0^T s_1(t) s_2^*(t+\tau) e^{(-j2\pi ft)} dt \tag{2-1}$$

where

s_1	received analytic signal from collector 1
s_2	received analytic signal from collector 2
	each with independent additive noise
τ	time lag parameters to be searched
f	frequency offset parameters to be searched

Each point in the CAF plane represents the magnitude of the correlation at a specific time and frequency offset. The highest degree of correlation occurs when coherent signal components are precisely aligned in both time and frequency. The values of the time and frequency offsets that maximize this function yield the “best” estimate of the signal’s TDOA and FDOA. Hence, the CAF is a three-dimensional surface with the coordinates of TDOA, FDOA, and magnitude. The peaks in the surface result from the correlation of coherent signal energy. Figure 2-1 shows a typical CAF result.

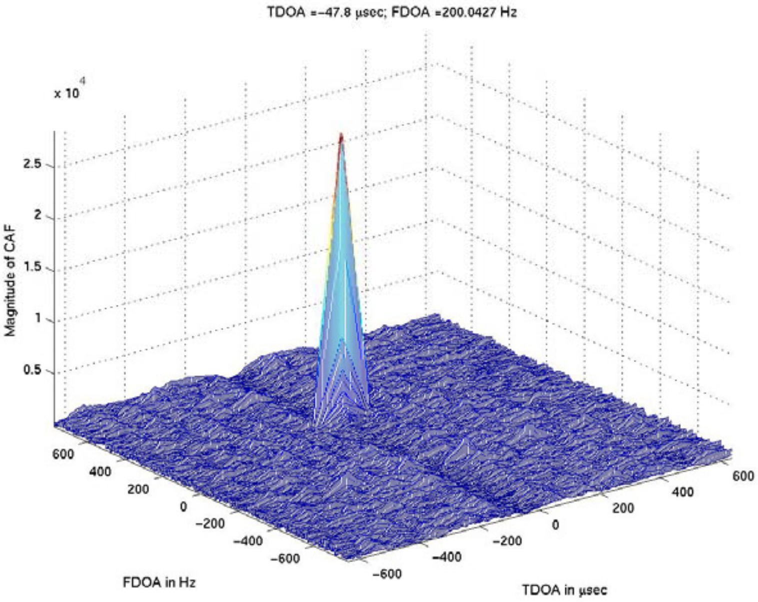


Figure 2-1: Example of a typical CAF result

The performance for the standard CAF algorithm is presented by Stein [2] and modified in Ulman and Geraniots [3]. This thesis presents only the results of their analysis. The standard deviation of the TDOA and FDOA measurements are given by Equations 2-2 and 2-3:

$$\sigma_{TDOA} = \frac{1}{\beta_s} \frac{1}{\sqrt{B_n T \gamma}} \tag{2-2}$$

$$\sigma_{FDOA} = \frac{0.55}{T} \frac{1}{\sqrt{B_n T \gamma}} \tag{2-3}$$

where

B_n noise bandwidth common to the two receivers

T integration time of the signal

β_s “rms Bandwidth” in the received signal spectrum

γ signal-to-noise ratio (SNR) given by (2-5)

$$\beta_s = 2\pi \left[\frac{\int_{-\infty}^{\infty} f^2 W_s(f)df}{\int_{-\infty}^{\infty} W_s(f)df} \right]^{1/2} \tag{2-4}$$

$$\frac{1}{\gamma} = \frac{1}{2}\left[\frac{1}{\gamma_1} + \frac{1}{\gamma_2} + \frac{2}{\gamma_1 \gamma_2} \right] \tag{2-5}$$

W_s is the signal's power spectral density and γ_i is the SNR of the i^{th} receiver in the receiver's noise bandwidth. For a signal with a constant envelope, such as a PSK signal, a good rule of thumb according to Stein [2] is $\beta_s \approx 1.8\, B_s$ as shown below:

$$\beta_s = \frac{\pi}{\sqrt{3}} B_s \tag{2-6}$$

where B_s is the signal RF bandwidth.

B. CAF MEASUREMENT PERFORMANCE

The CAF has been widely used in radar to illustrate the range and Doppler resolution properties of a radar waveform. These principles apply similarly to the CAF in terms of differential range and differential Doppler. There are two main signal factors that drive the performance of the CAF function; the signal bandwidth and the integration time of the signal. TDOA accuracy is most affected by the signal's bandwidth and FDOA accuracy is most effect by the integration time of the signal.

1. Effects of Signal Bandwidth on TDOA Measurement

As seen in Equation 2-2, the standard deviation is inversely proportional to the bandwidth of the signal. A large standard deviation implies low accuracy in estimating the position of the TDOA peak in the CAF surface, which in turn results from a wide TDOA peak in the CAF surfaces shown in Figures 2-2 and 2-3.

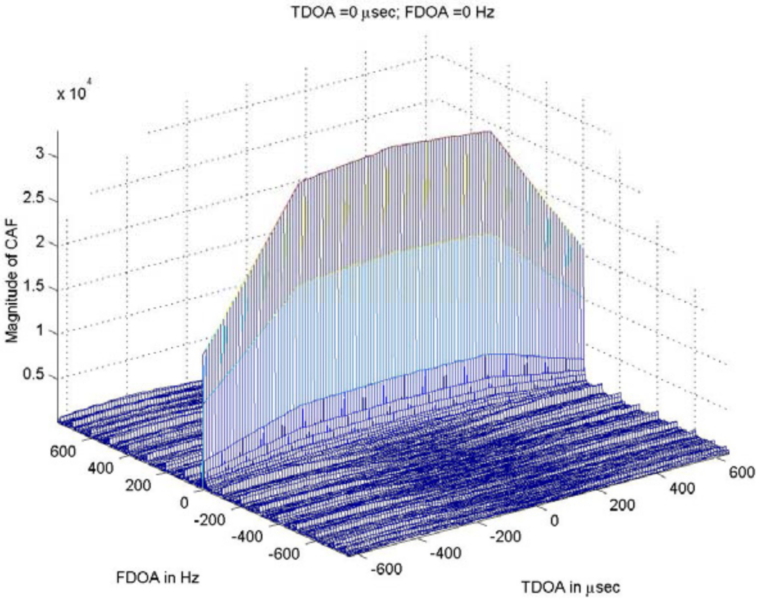


Figure 2-2: CAF peak with a narrow bandwidth signal

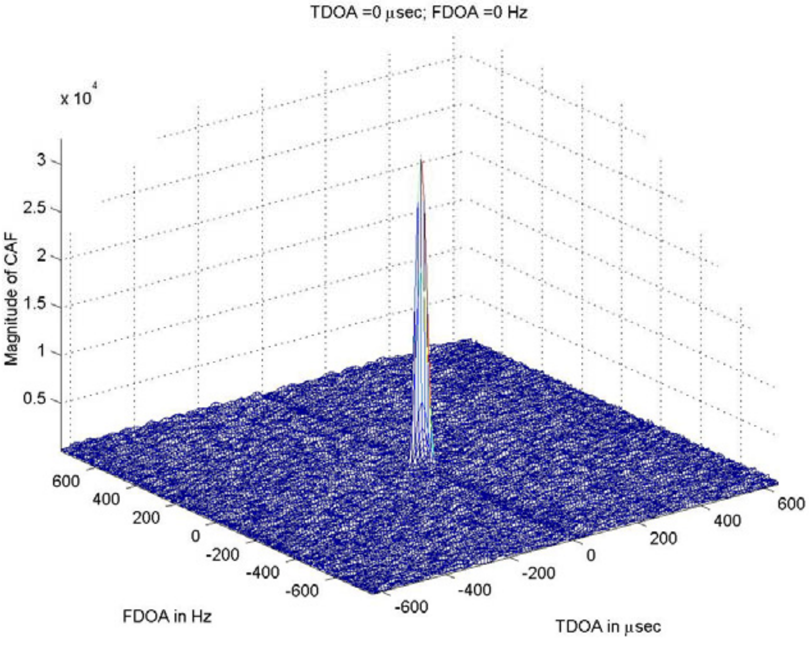


Figure 2-3: CAF peak with a wide bandwidth signal

In Figures 2-2 & 2-3, the sharpness of the primary peak along the TDOA axis is directly proportional to the bandwidth of the signal. Very narrow bandwidth signals, such as a CW tone, will have nearly constant amplitude along the TDOA axis in the CAF plane as seen in Figure 2-2, while a very wide signal will have a single narrow peak shown in Figure 2-3.

2. Effects of Integration Time on FDOA Measurement

The effect on the FDOA estimation accuracy is clearly seen in the CAF surfaces shown in Figures 2-4 & 2-5. As noted in Equation 2-3, the standard deviation for FDOA is inversely proportional to time. Short duration signals will have nearly constant amplitude along the FDOA axis in the CAF plane as seen in Figure 2-4. In Figure 2-5, a signal that was present for the entire snapshot gives a well-defined peak. For constant envelope signals, the correlation shape in the FDOA cross-section is a sin(x)/x shape with the width of the main lobe inversely proportional to the signal integration time.

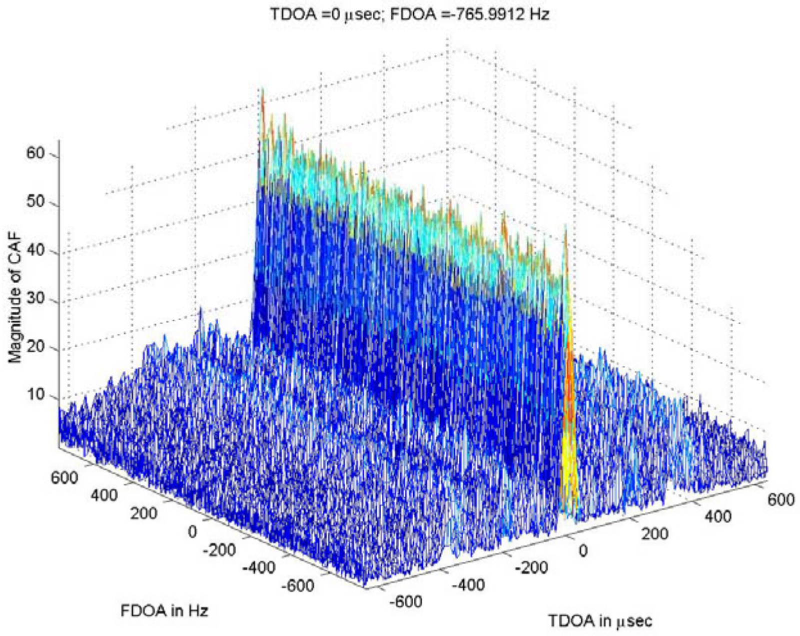


Figure 2-4: CAF peak with a short duration signal

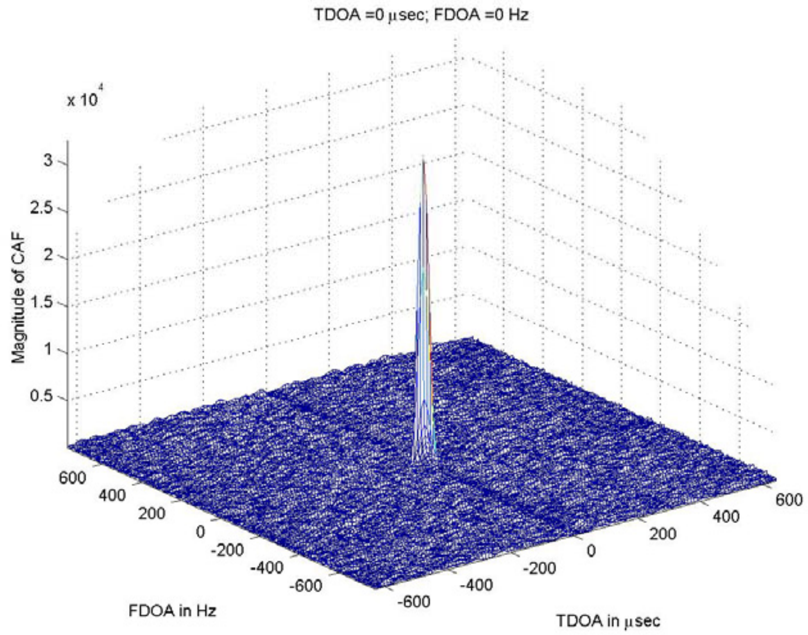


Figure 2-5: CAF peak with the signal duration as long as the snapshot

3. The Effects Caused by the FFT

The CAF is computed efficiently using the FFT. With this process, the basic increments of the time shifts are equal to the data sample period $\left(\frac{1}{F_s}\right)$. Computed FDOA increments are the reciprocal of the processing time window which may be extended beyond the actual data length of the snapshot for a finer representation of the surface. The $\frac{\sin(x)}{x}$ shape is a result of the FFT processing used to compute the CAF and it depends on the window used for the smoothing of the transform. For example, if a rectangular window were used, the 3 dB width of the main lobe would be $0.89\left(\frac{2\pi}{N}\right)$ where N is the number of samples, and the first side lobe is only 13 dB below the main lobe. Different windows can be used to trade between the width of the main lobe and the height of the side lobes.

The discussions on the FFT effects are of particular significance for the real world case of multiple co-channel interference. In practical cases, signals that share common spectra will be spatially separated. Spatial separation usually implies a corresponding separation in TDOA, FDOA or both. Thus, the basic limit to spatial resolution of multiple emitters is related directly to the TDOA and FDOA resolution of the signals in the ambiguity surfaces. This basic resolution is degraded by the presence of correlation side-lobes of a strong emitter over-shadowing the primary peak of a weaker emitter. Even disregarding the effects of the side-lobes, emitter separation over a great distance may not be resolved in a single CAF surface due to the left-right ambiguity inherent in a CAF based system.

As stated earlier, the TDOA measurement accuracy with conventional CAF processing is limited by the signal's bandwidth and, therefore, is not influenced by available collection parameters. FDOA measurement accuracy can be increased by longer collection duration up to the limit at which the excessive Doppler smearing and decorrelation due to higher order TDOA and FDOA derivatives and propagation effects become significant.

4. Collector Geometry Effects on TDOA and FDOA

The collector's geometry has a great effect on the TDOA and FDOA results. In general, it is best to place the collectors as far apart as possible within the limit that requires that both collectors have sufficient Signal to Noise Ratio (SNR) to process the CAF. Also, higher velocity vectors introduce more Doppler; therefore causing an increased FDOA.   But, this may require reduced snapshot durations to keep Doppler smearing to a minimum. It is important to realize that the TDOA is affected most by the separation of the collection platforms and the FDOA is most affected by the velocity vector of the platforms.

Figures 2-7 & 2-8 show the possible TDOA and FDOA results for a given geometry of two aircraft spaced 2 km apart in the x direction flying at an altitude of 6.7 km at 30 m/s velocity in the x direction. The signal is transmitted at 1 GHz in this example. The geometry is shown in Figure 2-6.

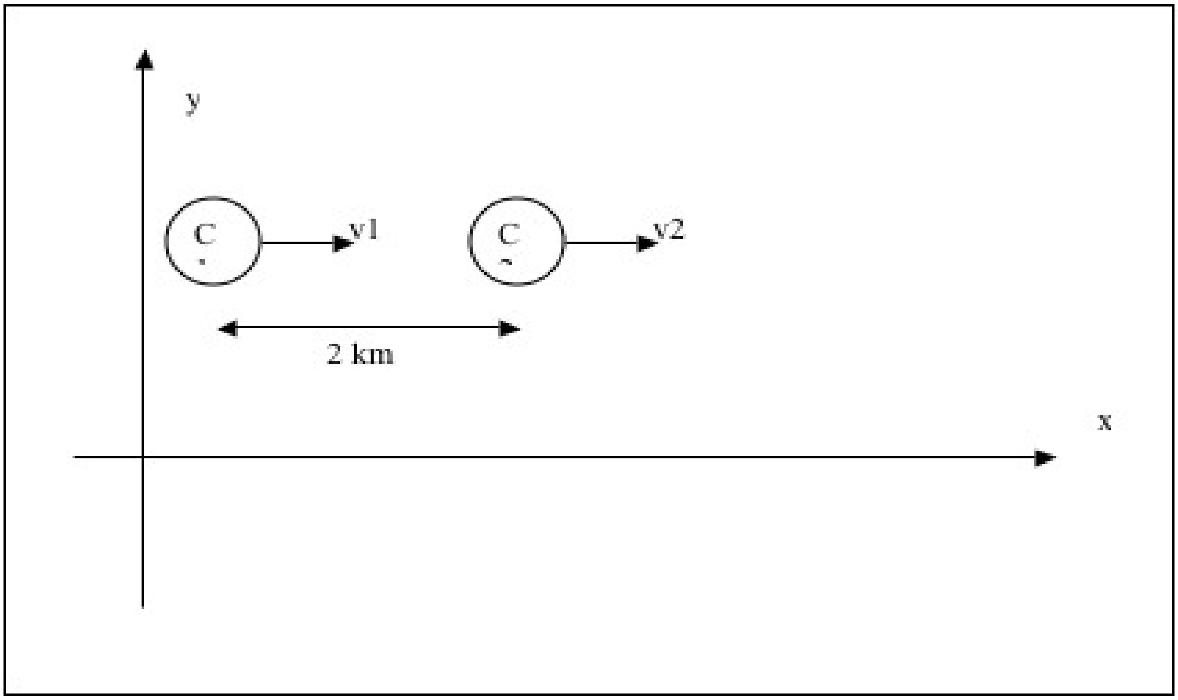


Figure 2-6: Collection geometry for Figures 2-7 and 2-8

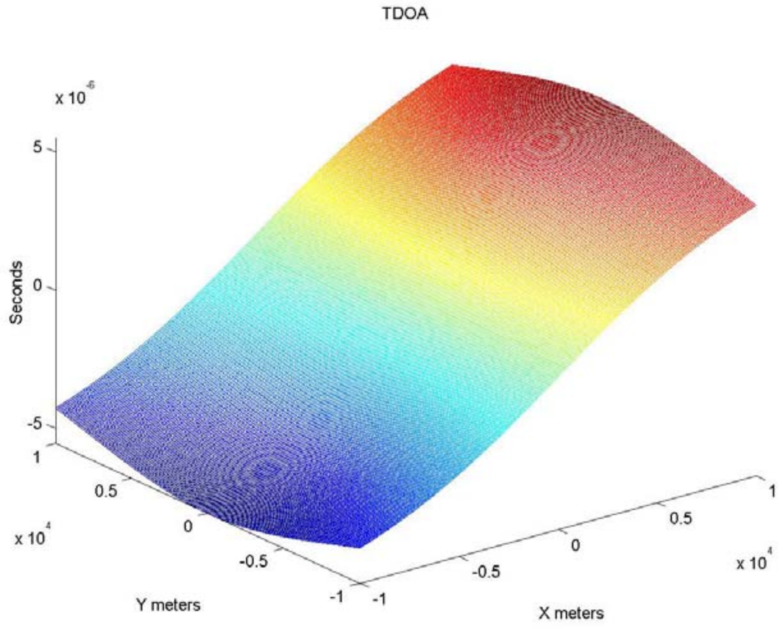


Figure 2-7: Possible TDOA isochrones values based on flight along x axis

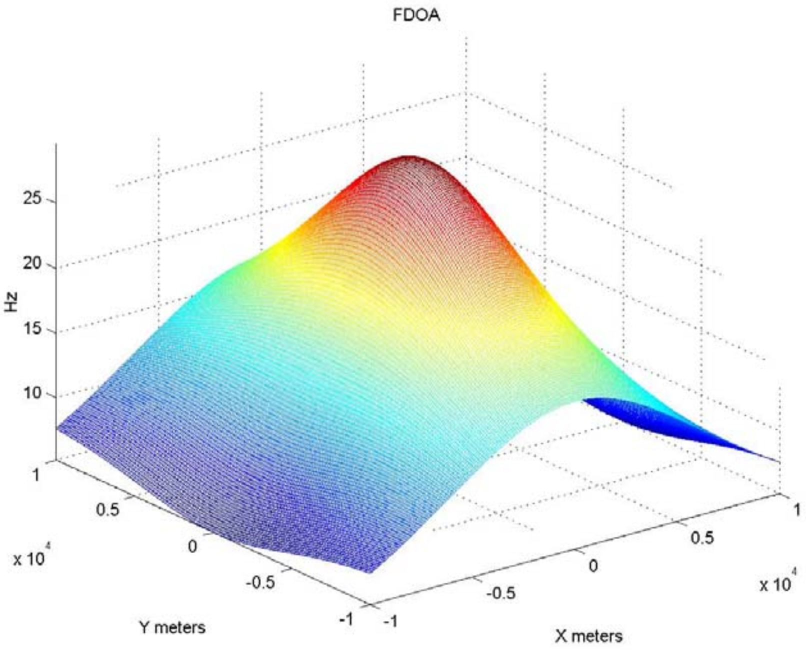


Figure 2-8: Possible FDOA isodops values based on flight along x axis

Figures 2-10 & 2-11 show a similar geometry with the same signal, but the aircraft are flying in the y direction at 30 m/s with the aircraft still separated by 2 km in the x direction. Figure 2-9 shows the geometry for this example.

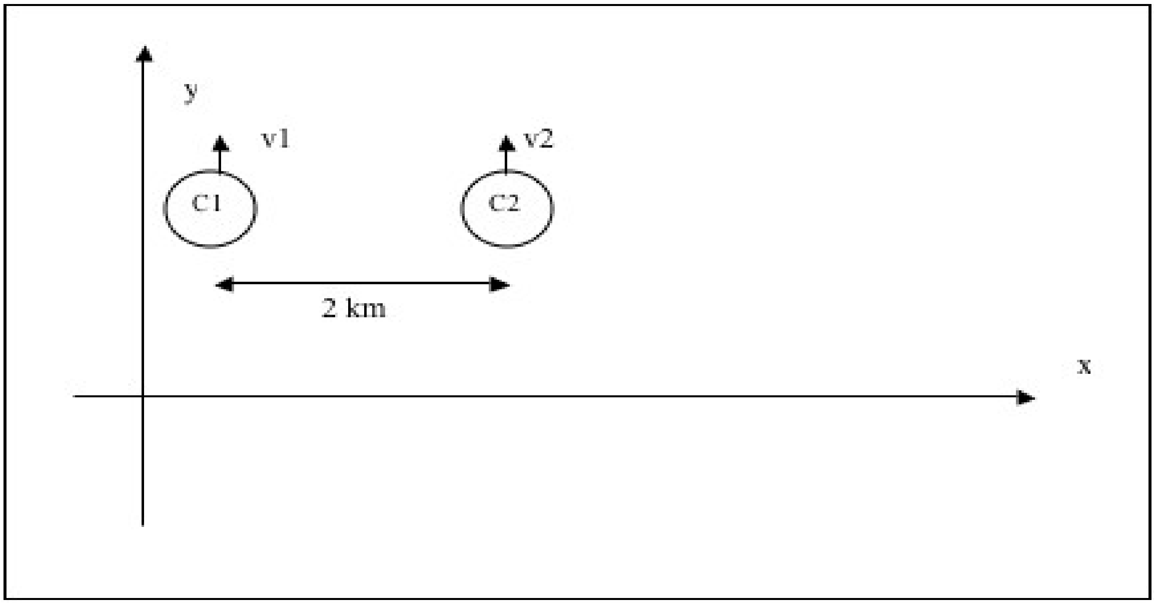


Figure 2-9: Collection geometry for Figurers 2-10 and 2-11

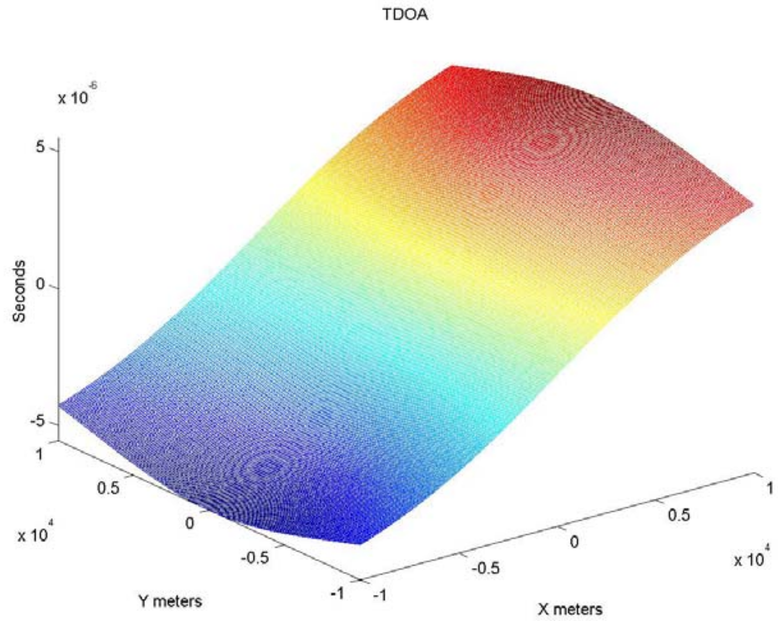


Figure 2-10: Possible TDOA isochrones values based on flight along y axis

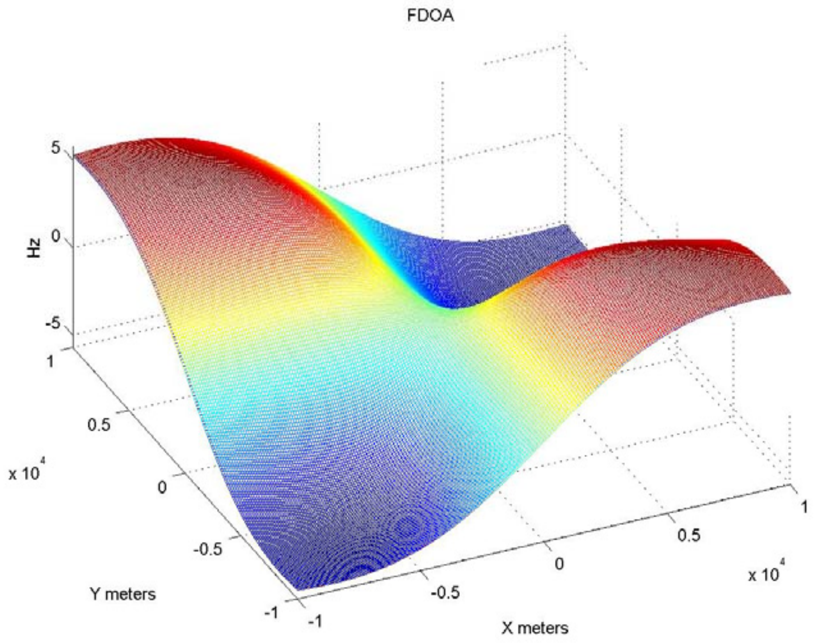


Figure 2-11: Possible FDOA isodops values based on flight along y axis

Note that the velocity vector does not change the TDOA isochrones. TDOA isochrones are dependent only on the sensor separation not on the signal's frequency or the platform's velocity vector. Therefore, the TDOA method can be used from stationary platforms if enough platforms with the correct geometry are used. Note that the FDOA isodops are dependent on the velocity vector and are not produced from a stationary platform.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TRADITIONAL TDOA/FDOA GEOLOCATION

Traditionally, a CAF is preformed on each Pre-D snapshot pair and the TDOA and FDOA are estimated by determining the peak or peaks in the CAF plane as shown in Figure 3-1. To determine the location estimate in *n* dimensions, *n* measurements are required. However, it is always useful to have an over-constrained problem to improve the accuracy of the solution. Once estimates are made of the FDOA and TDOA for a number of independent snapshots, the location can be determined. One of the most common methods used as the geolcation engine to solve this over-constrained problem is the Newton-Raphson method.

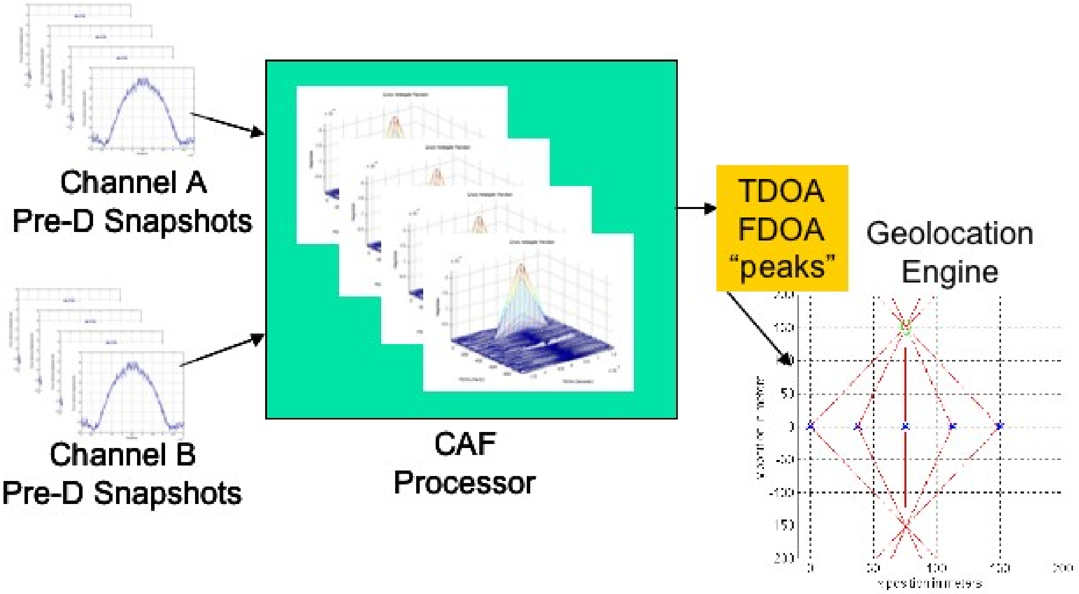


Figure 3-1: Traditional TDOA/FDOA Geolocation

A. NEWTON-RAPHSON METHOD

The Newton-Raphson method is a numerical approximation method to find the root of an equation. This iterative process follows a set guideline to approximate one or two roots, considering the functions, its derivative, and an initial x-value and y value. In the paper “Where is it?” [4] and the Stoner Memo: 129 “Dry Gulch Jake and the Goddess of the Desert” [5] both by Dr. J. Stoner as well as in Dr. H. Loomis’s paper “Geolocation of Electromagnetic Emitters” [6] the Newton-Raphson method is discussed for the location estimates from TDOA measurements. This method can be used with combined TDOA and FDOA measurements including their error sources such as the random and bias errors terms as well, but this simpler TDOA example is used to illustrate the method.

As a 2-dimensional example let's assume a pair of collection systems measure a TDOA *m* from an emitter at $\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$ at time t_j. *m* is a function of position $\mathbf{p}$ and time *t.*

$$m_j = f(\mathbf{p}, t_j) \tag{3-1}$$

In vector form for multiple *k* collections this becomes:

$$\mathbf{m} = f(\mathbf{p}, t) \tag{3-2}$$

The problem is that we have $\mathbf{m}$ and we want $\mathbf{p}$. We have k observations giving us an over constrained set of equations to solve for $\mathbf{p}$. This seems like a straightforward least squares problem where we can invert this relationship and easily solve for $\mathbf{p}$, but $f(:,t)$ is non-linear and hence non invertible. This is where the Newton-Raphson method is used to estimate the location. The first step is to linearize $f(:,t)$ by using the first few terms of the Taylor series approximation of the function $f(:,t)$ in the vicinity of a suspected root $\mathbf{p}$ based on the value of $\mathbf{m}$ at a estimate of the position $\mathbf{p}$, call it $\mathbf{p}_0$.

$$\mathbf{m} = f(\mathbf{p}_0, t) + \left.\frac{\delta f}{\delta \mathbf{p}}\right|_{\mathbf{p}_0} \cdot (\mathbf{p} - \mathbf{p}_0) + \text{higher order terms} \tag{3-3}$$

So, now $\mathbf{m}_0$ can be calculated based on the guess of $\mathbf{p}_0$, and the higher order terms can be dropped leaving:

$$\mathbf{m} - \overbrace{\mathbf{m}_0}^{\text{calculated}} \approx \left.\frac{\delta \mathbf{f}}{\delta \mathbf{p}}\right|_{\mathbf{p}_0} \cdot \left(\underbrace{\mathbf{p}}_{\text{estimated}} - \overbrace{\mathbf{p}_0}^{\text{guess}} \right) \tag{3-4}$$

or

$$\underbrace{\delta\mathbf{m}}_{kx1}=\underbrace{\mathbf{A}}_{kx2}\underbrace{\delta\mathbf{p}}_{2x1} \tag{3-5}$$

where **A** is a non-invertible matrix of the partials with respect to *x* and *y* of the function $\mathbf{f}(\mathbf{p},t)$.

The best way to determine the partials contained in **A** is to work in the vector space and take the gradient of TDOA function.

$$\nabla f=\frac{\delta f}{\delta x}\boldsymbol{x}+\frac{\delta f}{\delta y}\boldsymbol{y} \tag{3-6}$$

The positions and the velocities of the platforms are denoted by $\mathbf{p}_1, \mathbf{p_2} \in$ R^2 and $\mathbf{v}_1, \mathbf{v_2} \in$ R^2, respectively. The unknown location is **p** and it is assumed that its velocity is zero. The unit vectors from the unknown emitter's location to the two platforms are given by

$$\mathbf{u}_i=\frac{\mathbf{p}_i-\mathbf{p}}{\left\|\mathbf{p}_i-\mathbf{p}\right\|} \tag{3-7}$$

The TDOA $\tau(\mathbf{p})$ and FDOA $\upsilon(\mathbf{p})$ are given by Equations 3-8 and 3-9:

$$\tau(\mathbf{p})=\frac{1}{c}\left(\left\|\mathbf{p}_2-\mathbf{p}\right\|-\left\|\mathbf{p}_1-\mathbf{p}\right\|\right) \tag{3-8}$$

$$\upsilon(\mathbf{p})=\frac{f_0}{c}\left[\mathbf{v}_2^T\mathbf{u}_2-\mathbf{v}_1^T\mathbf{u}_1\right] \tag{3-9}$$

where *c* is the speed of light and f_0 is the emitters center frequency. Their gradients are

$$\nabla\tau(\mathbf{p})=-\frac{1}{c}\left(\mathbf{u}_2-\mathbf{u}_1\right) \tag{3-10}$$

$$\nabla \upsilon(\mathbf{p}) = \frac{f_0}{c}\left(\frac{\left[\mathbf{I}-\mathbf{u}_2\mathbf{u}_2^T\right]\mathbf{v}_2}{\left\|\mathbf{p}_2-\mathbf{p}\right\|} - \frac{\left[\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T\right]\mathbf{v}_1}{\left\|\mathbf{p}_1-\mathbf{p}\right\|} \right) \tag{3-11}$$

From the gradients of the TDOA and FDOA equations, a combined TDOA/FDOA partial derivative matrix **A** can be built.

$$\mathbf{A}_i = \begin{bmatrix} \frac{\delta\tau}{\delta x} & \frac{\delta\tau}{\delta y} \\ \frac{\delta\upsilon}{\delta x} & \frac{\delta\upsilon}{\delta y} \end{bmatrix} i = 1..k \tag{3-12}$$

or

$$\mathbf{A}_i = \begin{bmatrix} \nabla \tau(\mathbf{p}) \\ \nabla \upsilon(\mathbf{p}) \end{bmatrix} \tag{3-13}$$

Now, let's return to the TDOA-only example. Again, we have the problem of not being able to invert **A**. We could use the least-squares method and use the pseudo-inverse of **A** by calculating $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ or we can use a conditioning matrix **W** to calculate a weighted pseudo-inverse of **A**. If we know that the system has different error sources for the TDOA and FDOA measurements, the best approach is to use the weighted pseudo-inverse of **A**. This approach is called the weighted least squares (WLS) solution. By using the WLS method we can account for variations in the measurement quality. Dr. Michael Price gives an excellent refresher in his memos "Covariance and Information Matrices: A Primer" [7] and "Least Squares Geolocation Data Combining – a Summary" [8]. The pseudo inverse of **A** is also known as the parameter covariance matrix **V** given by Equation (3-14)

$$\mathbf{V} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1} \tag{3-14}$$

Inserting this into Equation 3-5 and solving for $\delta\mathbf{p}$, the WLS estimate for $\delta\hat{\mathbf{p}}$ is given by Equation 3-15.

$$\delta\hat{\mathbf{p}} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\delta\mathbf{m} \tag{3-15}$$

1. The Weighting Matrix

The weighting or conditioning matrix **W** is chosen to account for the differences in quality of the observations and is the inverse of the covariance matrix **R**.

$$\mathbf{W} = \mathbf{R}^{-1} \tag{3-16}$$

where **R** is

$$\mathbf{R} = \begin{bmatrix} \sigma_0^2 & 0 & \dots & 0 \\ 0 & \sigma_1^2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & \sigma_{k-1}^2 \end{bmatrix} \tag{3-17}$$

In this simple two-dimensional example we can use Equation 3-18 for the weighting matrix **W**.

$$\mathbf{W} = \begin{bmatrix} 1/\sigma_1^2 & 0 \\ 0 & 1/\sigma_2^2 \end{bmatrix} \tag{3-18}$$

where $\sigma_1{}^2$ and $\sigma_2{}^2$ are the timing error variances and are, to the first approximation, the root sum square (RSS) of the random and bias errors. Now we have everything we need to bring an algorithm together to make successively better estimates of the emitter's location $\mathbf{p}_0$.

The algorithm shown is the one that Dr. H. Loomis used in his paper "Geolocation of Electromagnetic Emitters" [6].

Algorithm

```
1. Make an estimate of the emitter location $\mathbf{p}_0$.
2. $i \leftarrow 0$
3. Compute $\mathbf{m}_i = \mathbf{f}\left(\mathbf{p}_i\right)$
4. $\delta\mathbf{m} = \mathbf{m} - \mathbf{m}_i = \mathbf{A} \cdot \left(\mathbf{p}_{i+1} - \mathbf{p}_i\right) = \mathbf{A}\delta\mathbf{p}_{i+1}$ where $\left(\mathbf{m} - \mathbf{m}_i\right)$ are the residuals.
4a. Exit if residuals are small enough.
5. $\delta\mathbf{p}_i = \mathbf{p}_{i+1} - \mathbf{p}_i = \left[\mathbf{A}^T\mathbf{W}\mathbf{A}\right]^{-1}\left[\mathbf{A}^T\mathbf{A}\right]\delta\mathbf{m}$
6. Calculate $\mathbf{p}_{i+1} = \mathbf{p}_i + \delta\mathbf{p}_i$
7. $i \leftarrow i+1$
8. $\rightarrow 3$
```

B. THE CONFIDENCE ELLIPSE

After calculating the estimate for the emitter's position and the variances that are associated with the estimate, the confidence ellipse can be calculated. Calculating the 95% confidence ellipse is a standard technique and has been explained by Clark [9] and Daniels [10]. Only the results will be summarized here. If we look at the parameter covariance matrix **V** where:

$$\mathbf{V} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1} \tag{3-19}$$

from Equation (3-14), it can now be written in the form:

$$\mathbf{V} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \tag{3-20}$$

The diagonal terms, σ_x^2 and σ_y^2, are the variances and ρ is the correlation coefficient for the parameters x and y. The off-diagonal term, $\rho\sigma_x\sigma_y$, is the covariance for x and y. These terms are now used in the calculation of the semi-major (a) and semi-minor (b) axes and orientation (θ) of the confidence ellipse.

$$(\sigma_a + \sigma_b)^2 = \sigma_x^2 + \sigma_y^2 + 2\sigma_x\sigma_y(1-\rho^2) \qquad (3\text{-}21)$$

$$(\sigma_a - \sigma_b)^2 = \sigma_x^2 + \sigma_y^2 - 2\sigma_x\sigma_y(1-\rho^2) \qquad (3\text{-}22)$$

where σ_a^2 and σ_b^2 are the variances of the contour error ellipse semi-major and semi-minor axes respectively. Since we have assumed that the errors are normally distributed, have zero means, and are independent, the axes' solution can be written as a sum of squares of two stochastically independent variables and their result is a chi-square distribution as found in Papoulis [11]. Therefore, we can write

$$\left(\frac{a}{\sigma_a}\right)^2 + \left(\frac{b}{\sigma_b}\right)^2 = \chi^2 \qquad (3\text{-}23)$$

The quantity χ^2 can be found in statistical tables for the chi-square distribution. For a 95% contour ellipse, the result is $\chi^2 = 5.991$ so

$$\left(\frac{a}{\sigma_a}\right)^2 + \left(\frac{b}{\sigma_b}\right)^2 = 5.991 \qquad (3\text{-}24)$$

Rearranging and solving explicitly for a and b we get Equations 3-25 and 3-26:

$$a = 2.448\sigma_a \qquad (3\text{-}25)$$

$$b = 2.448\sigma_b \qquad (3\text{-}26)$$

The ellipse orientation is given by equation 3-27:

$$\theta = \frac{1}{2}\tan^{-1}\left\{\frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 - \sigma_y^2}\right\} \qquad (3\text{-}27)$$

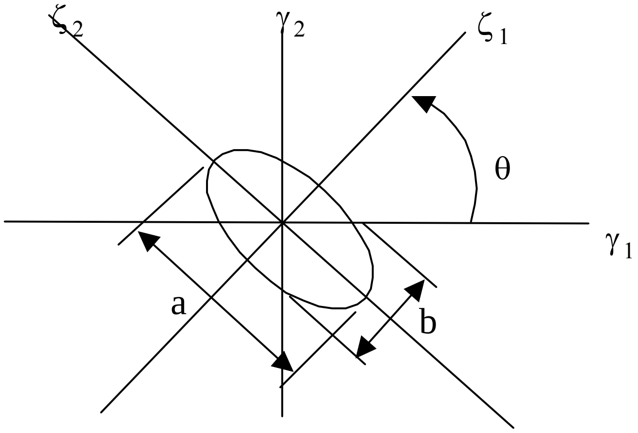


Figure 3-2: Illustration of the confidence ellipse.

The traditional geolocation method discussed in this chapter is well understood and has been used for many years. The major draw back of this method is that it depends on the accuracy of the peak determination in the CAF surface. If the CAF is computed for a co-channel environment where there are multiple peaks in the CAF surface, it is impossible to determine the correct peak from among the multiple peaks caused by signal mixing within the correlation process. In the next Chapter, the CAF-Map method is discussed; this method skips the part of the traditional geolocation process where the peak in the CAF surface is determined and only the value of the TDOA and FDOA are passed to the geolocation engine. Instead, the entire CAF surface is "mapped" to the Earth's surface, thus eliminating one of the major error sources in the traditional method.

IV. THE CAF-MAP METHOD

The task of geolocation of multiple simultaneous co-channel emitters consists primarily of identifying and associating primary correlation peaks across multiple CAF surfaces and across multiple independent collections. Multiple primary peaks are often indistinguishable from the secondary correlation side lobes and cross-modulation artifacts of a single strong emitter.

The processing technique proposed in this thesis relies on the fundamental principle that primary-correlation peaks for stationary emitters will be perfectly consistent for all CAF surfaces. In effect, all available CAF surfaces are mapped and combined in a common geographic frame which results an image analogous to radio imaging. The apparent position of spurious artifacts, secondary side lobes, and the left-right images will lack the consistency of the true peaks due to the varied geometry and dynamics of the collection platforms. This method eliminates the step were the TDOA and FDOA values from several snapshots are converted to an estimate of the location using the Newton Raphson method. The CAF-Map method simply maps the TDOA and FDOA values in the CAF surface directly to an X, Y coordinate system. Figure 4-1 shows the steps of the CAF-Map method.

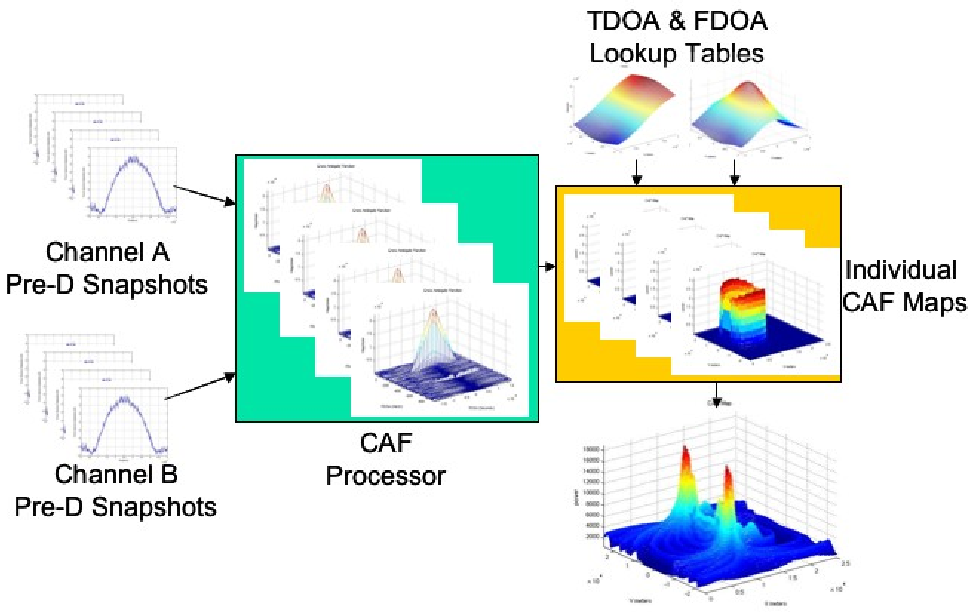


Figure 4-1: The CAF-Map method

The method used in this approach follows:

1. Calculate the theoretical TDOA and FDOA offsets for points on the X, Y grid for the current snapshot's geographic coverage to create a lookup table of FDOA(s) and TDOA(s).
2. Calculate the normal CAF surface for current snapshot.
3. Use the lookup table in step A to "map" the amplitude of the CAF in step B to a new X, Y surface.
4. Repeat 1 though 3 and sum maps over a number of snapshots.

This is the method, called the CAF-MAP method [1], which was proposed by Mr. Al Buczek of the Naval Research Laboratory in the early 1990's. In this method, the entire geographic coverage area's TDOA(s) and FDOA(s) are computed to form a lookup table for each snapshot. Then, each snapshot's CAF is computed over the range of the expected TDOA(s) and FDOA(s). Once the CAF(s) are computed for each snapshot, a geographic MAP can be formed using the lookup tables to "map" the CAF to the ground. Then each "map" is summed over a common geographic area to provide a RF energy map of the area for the collected frequency. This method produces a geographic "image" of the geolocated energy instead of the traditional map with an error ellipse. The master script that calls the functions required for this method is called the "caf_map.m" function.

The function "caf_map.m," listed in Appendix A, is a function written in MATLAB® that computes the CAF and the associated CAF-Map based upon the input signals and geographic area. The function is invoked on the command line of the form:

[map,PtempX,PtempY]=caf_map(S2,S1,Fo,Fs,dm,Pe1,Pe2,Pc1,Vc1,Pc2,Vc2);

The input arguments *S2* and *S1* are the collected analytic signal snapshots from each of the two platforms. The input arguments *Fo* and *Fs* are the carrier frequency of the intercepted signal and the sampling rate in Hz of the receivers' digitizer respectively. The input argument *dm* is the desired *x* and *y* resolution of the CAF-Map image. The input arguments *Pe1* and *Pe2* describe the area to calculate CAF-Map image. These arguments are two dimensional [*x, y*] in meters. The north-south direction is the *y* argument while the east-west is the x argument. It is assumed that the grid points are on the surface of a flat earth and that the altitude is zero meters; however, with small changes to this function, Digital Terrain Elevation Data (DTED) could be used to improve results. The input arguments *Pc1, Vc1, Pc2* and *Vc2* describe the collector's position and velocity vectors at the middle of the snapshot. *Pc1* and *Pc2* are each three dimensional entries and are in [*x, y, z*] form. The collector's altitude is in the z direction. All three arguments are in meters. *Vc1* and *Vc2* are each three dimensional entries and are in [*Vx, Vy, Vz*] form. These describe each collector's velocity vector and are in meters/seconds.

This function calls three other functions written in MATLAB®, tdoa_fdoa_grid3D.m, CAF_peak.m, and map_tdoa_fdoa.m. The function tdoa_fdoa_grid3D.m calculates the expected TDOAs and FDOAs for a geographic area. CAF_peak.m calculates the CAF plane, and map_tdoa_fdoa.m maps the TDOA's and FDOA's amplitude and phase found in the CAF plane to an *x, y* location on the map. Keeping the amplitude and phase information allowed experimenting with coherently combining the snapshots. This proved to be unsatisfactory and as seen in section E of this chapter, the magnitude of each snap-shot was used in combining the snapshots to form the energy maps.

These functions produce two plots one shows the CAF plane and the other shows the CAF-Map for the two input signals and collector geometry. Figures 4-2 and 4-3 show examples of the CAF plane and the CAF-Map.

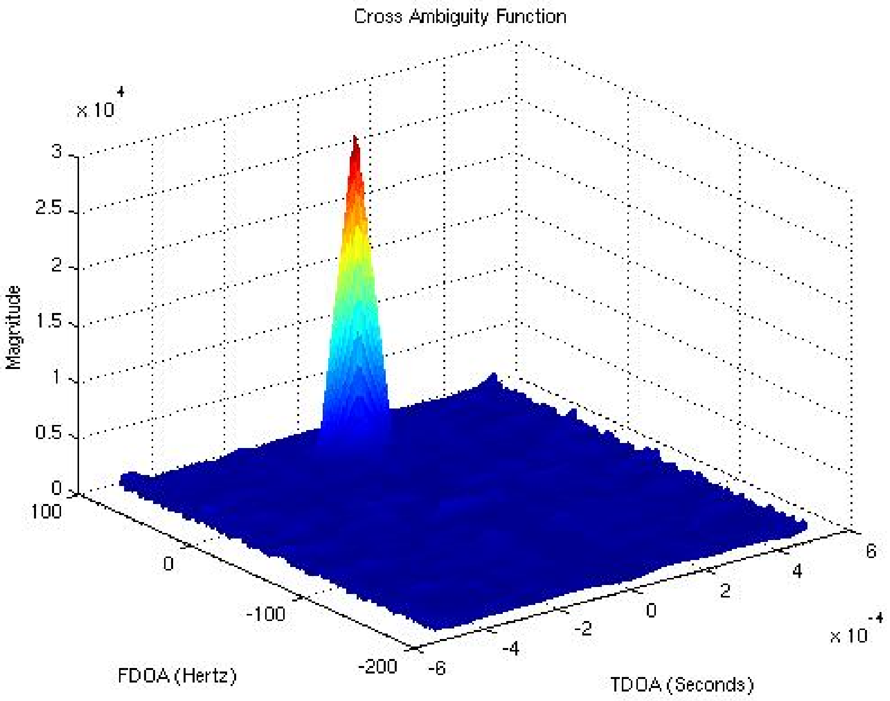


Figure 4-2: Example of a CAF plane generated by the 'caf_map.m' function

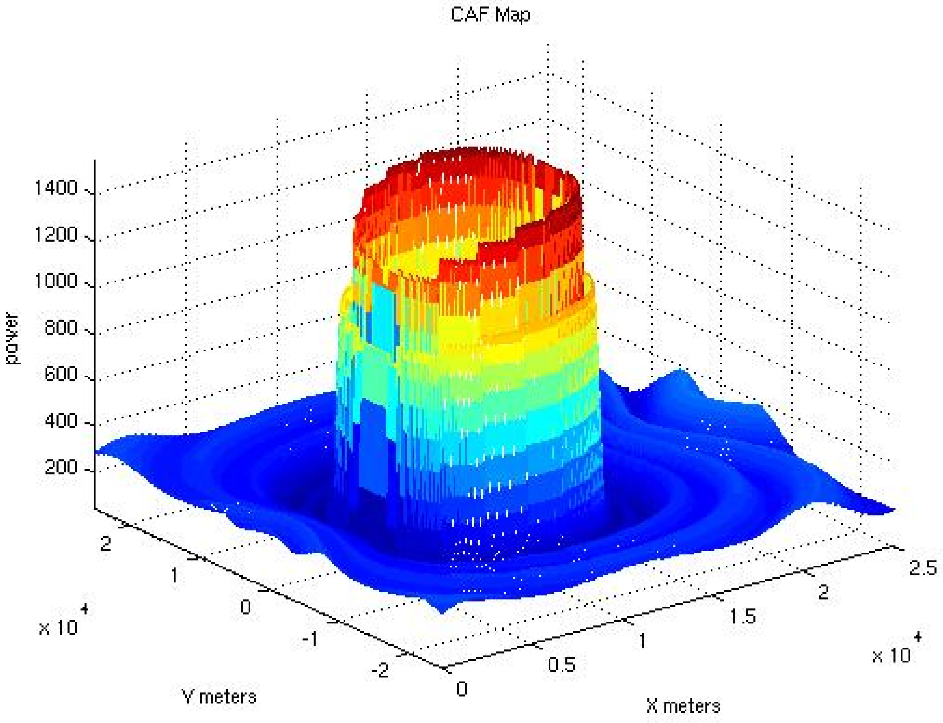


Figure 4-3: Example of a CAF-Map generated by the 'caf_map.m' function

A. TDOA & FDOA LOOKUP TABLES

For each snapshot, the lookup tables are computed for the theoretical TDOA(s) and FDOA(s) over a common geographic area. Figure 4-4 shows an example of a two dimensional Emitter-Collector geometry. To create the lookup tables the theoretical TDOA and FDOA are calculated for each grid point. The positions x_E and y_E for the grid location are changed to fill out the table. The MATLAB® function that calculates the TDOA and FDOA look up tables is called "tdoa_fdoa_grid3D.m".

The function "tdoa_fdoa_grid3D.m," listed in Appendix A, is a function written in MATLAB® that computes the theoretical TDOA and FDOA for each grid point in a user defined area. The function is called from the main program called "caf_map.m" or it is invoked on the command line of the form:

[tdoa_grid, fdoa_grid, indexX, indexY] =
tdoa_fdoa_grid3D(Pc1,Vc1,Pc2,Vc2,Pe1,Pe2,f0,dm);

The input arguments *Pc1, Vc1, Pc2* and *Vc2* describe the collector's position and velocity vectors at the middle of the snapshot. *Pc1* and *Pc2* are each three dimensional entries and are in [*x, y, z*] form. The north-south direction is the y argument with east-west being the x argument. The collector's altitude is in the z direction. All three arguments are in meters. *Vc1* and *Vc2* are each three dimensional entries and are in [*Vx, Vy, Vz*] form. These describe each collector's velocity vector and are in meter/seconds. The input arguments *Pe1* and *Pe2* describe the area to calculate the TDOA(s) and FDOA(s). These arguments are each two dimensional [*x, y*] in meters. It is assumed that the grid points are on the surface of a flat earth and that the altitude is zero meters. The input argument f_0 is the carrier frequency of the intercepted signal in Hz. The last input argument is *dm*. This argument controls the resolution of the grid points in meters.

The output variables tdoa_grid and fdoa_grid are matrices that contain the TDOA(s) and FDOA(s) calculated for each grid point. The output variables indexX and indexY are the indices for the two matrices tdoa_grid and fdoa_grid.

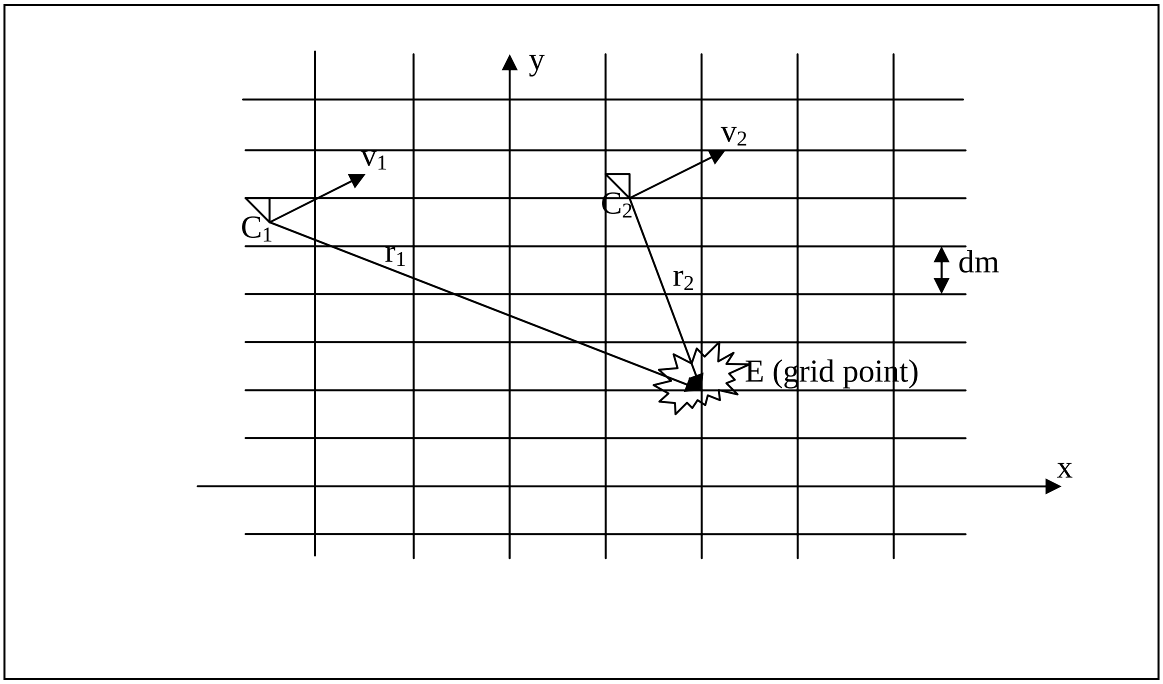


Figure 4-4: 2-D Emitter-Collector Geometry

In Figure 4-4, C$_1$, C$_2$, and E represent collector one, collector two, and the emitter (grid point) while $\mathbf{r_1}$ and $\mathbf{r_2}$ represent the position vectors from collectors to the emitter. The velocities vectors for each collector are represented by $\mathbf{v}_1$ and $\mathbf{v}_2$.

1. Calculating Theoretical TDOA(s)

The Time Difference of Arrival (TDOA) is simply the difference in time for the signal to propagate to one collector vice the other taken with respect to the second collector of a two-collector system. Equation (4-1) is the basic TDOA equation.

$$TDOA = \frac{|\mathbf{r}_2| - |\mathbf{r}_1|}{c} \tag{4-1}$$

where c is equal to the speed of light.

The vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are the differences between the x and y coordinates of the emitter or grid point for our function and the collectors. The three dimensional versions are shown in Equation (4-2):

$$\mathbf{r}_1 = \begin{bmatrix} x_E - x_{C_1} \\ y_E - y_{C_1} \\ z_E - z_{C_1} \end{bmatrix} \;\&\; \mathbf{r}_2 = \begin{bmatrix} x_E - x_{C_2} \\ y_E - y_{C_2} \\ z_E - z_{C_2} \end{bmatrix} \tag{4-2}$$

The distance, or norm, of the vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are determined by using the Pythagorean Theorem. This gives us the familiar form for the TDOA equation in three dimensions shown below:

$$TDOA = \frac{1}{c}\begin{bmatrix} \left[\sqrt{(x_E - x_{C_2})^2 + (y_E - y_{C_2})^2 + (z_E - z_{C_2})^2}\,\right] \\ -\left[\sqrt{(x_E - x_{C_1})^2 + (y_E - y_{C_1})^2 + (z_E - z_{C_1})^2}\,\right] \end{bmatrix} \tag{4-3}$$

The MATLAB® code to calculate this is:

```
tdoa_grid(i,j) = (norm(gridP - Pc2) - norm(gridP - Pc1)) / c;
```

where *gridP* is the grid point that the TDOA is being computed. *Pc1* and *Pc2* are the positions of the collectors. The MATLAB® code is similar to Equation (4-1). A graphical example of a TDOA lookup table is shown in Figure 4-5, the z-axis represents the TDOA value for each grid point.

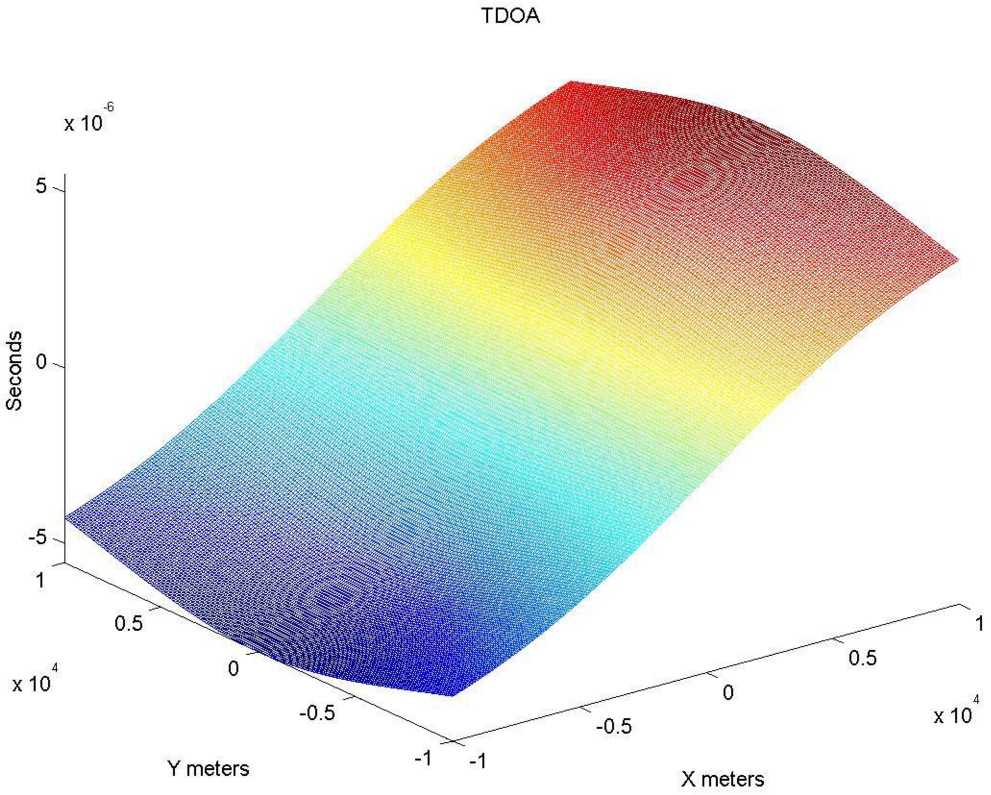


Figure 4-5: Example of a TDOA lookup table

2. Calculating Theoretical FDOA(s)

The FDOA between two collectors is simply the differences between the Doppler shifts that each collector intercepts. Using the geometry shown in Figure 4-4, the Doppler shift between one of the collectors and the emitter (grid point) is:

$$f_d = \frac{f_0}{c} v \tag{4-4}$$

where f_0 is the emitter's carrier frequency, c is the speed of light, and v is the velocity of closure between the collector and the emitter. This can be found by dividing the dot product between the vectors $\boldsymbol{v}$ and $\boldsymbol{r}$, by the norm of $\boldsymbol{r}$ as shown in Equation (4-5).

$$v = \frac{\mathbf{v} \bullet \mathbf{r}}{|\mathbf{r}|} \tag{4-5}$$

The vector $\boldsymbol{v}$ describes the relative velocity components in the *x*, *y*, and *z* directions shown in Equation (4-6).

$$\mathbf{v} = \begin{bmatrix} v_{E_x} - v_{C_x} \\ v_{E_y} - v_{C_y} \\ v_{E_z} - v_{C_z} \end{bmatrix} \tag{4-6}$$

Because we are calculating the FDOA for a grid point, it is assumed that the velocity of the emitter is zero. This reduces Equation 4-6 to the following:

$$\mathbf{v} = -\begin{bmatrix} v_{C_X} \\ v_{C_Y} \\ v_{C_Z} \end{bmatrix} \tag{4-7}$$

Substituting Equation (4-7) and (4-2) into (4-4) and a little simplification gives us the following for the Doppler equation:

$$f_d = \frac{-f_0}{c}\left[\frac{v_{C_X}\left(x_E - x_C\right) + v_{C_Y}\left(y_E - y_C\right) + v_{C_Z}\left(z_E - z_C\right)}{\sqrt{(x_E - x_C)^2 + (y_E - y_C)^2 + (z_E - z_C)^2}}\right] \tag{4-8}$$

The MATLAB® code to calculate the Doppler for each platform is:

*doppler1(i,j) = f0/c * dot(-Vc1, gridP-Pc1) / norm(gridP - Pc1);*

*doppler2(i,j) = f0/c * dot(-Vc2, gridP-Pc2) / norm(gridP - Pc2);*

where *Vc1* and *Vc2* are the velocity vectors for the collecting platforms, *Pc1* and *Pc1* are the collector's positions. Note that this is very similar to Equation (4-5).

FDOA is the difference in Doppler. Once the Doppler is calculated for each collector the difference is taken.

$$FDOA = f_{d2} - f_{d1} \tag{4-9}$$

or

fdoa_grid(i,j) = doppler1(i,j) - doppler2(i,j);

in MATLAB®.

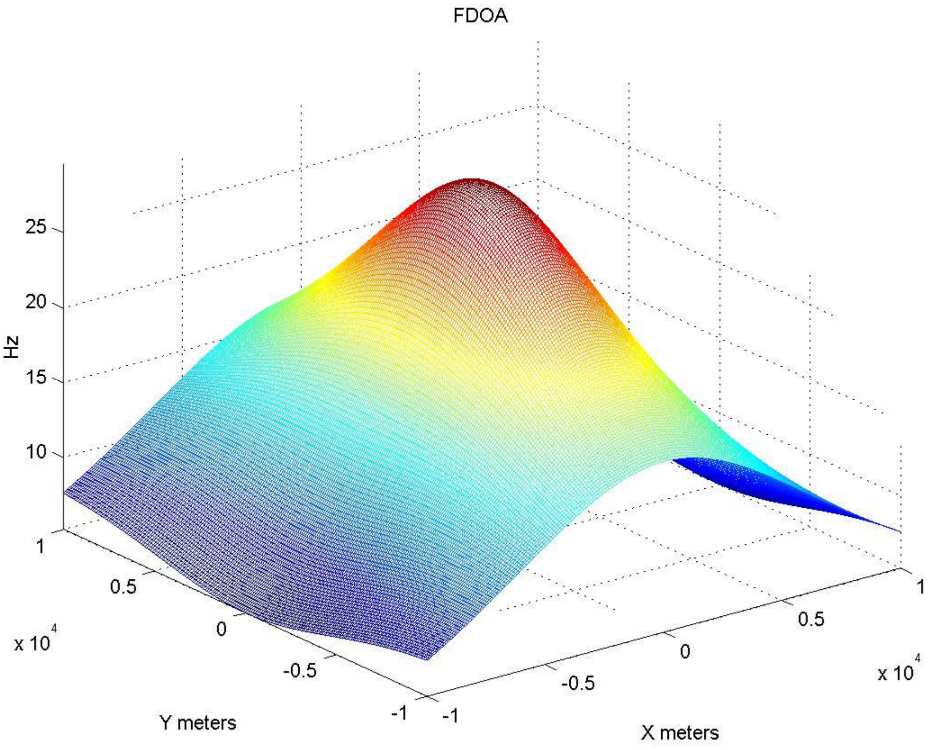


Figure 4-6: Example of a FDOA lookup table

Figure 4-6 shows an example of a graphical FDOA lookup table; in this figure the z axis shows the theoretical FDOA value for each grid point.

B. CALCULATE THE CAF PLANE

From the lookup table, the minimum and maximum of the expected TDOA(s) and FDOA(s) for the geographic region covered by the CAF-MAP are fed into the CAF processor along with the collected-signal snapshots. The CAF engine that is used for this thesis is a slightly modified version of one that was developed by LCDR Joe J. Johnson for his Masters Thesis at the Naval Postgraduate School completed Sept. 2001[12]. His engine's MATLAB® function is called "CAF_peak.m" but the CAF-Map technique is not limited to this particular CAF engine. This engine was chosen due to its ease of use and since it allowed the input TDOA and FDOA range to be controlled. However, this function did require some modifications to improve resolution and add additional output arguments needed to demonstrate the CAF-Map method.

The function CAF_peak.m, listed in Appendix A, is a function written in MATLAB® that computes the CAF surface by calculating the cross correlation between two signals in both time and frequency offsets. The function is called from the main program called “caf_map.m” or it is invoked on the command line of the form:

[TDOA, FDOA, MaxAmb, Amb, TauValues,FreqValues] =
CAF_peak(S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi, Fs, intp)

The input arguments *S1* and *S2* are the two input signal vectors in analytic signal format. The arguments *Tau_Lo*, and *Tau_Hi* represent the lowest and highest value of TDOA expected over the coverage area expressed as discrete time delays in samples for which to compute the CAF surface. Likewise, *Freq_Lo*, and *Freq_Hi* represent the lowest and highest FDOA expected for the coverage area expressed as digital frequencies. The input argument *Fs* is the sampling frequency of the input arguments *S1* and *S2*. The last input argument is *intp*, this argument controls the interpolation of the CAF plane. A value of ‘0’ turns off the interpolation and a value of intp other than ‘0’ turns on the interpolation. The value of *intp* other than ‘0’ controls the number of points that the CAF plane is interpolated by. During this thesis, a value of 10 was sufficient for the *intp* value. The output arguments *TDOA* and *FDOA* are the TDOA and FDOA calculated for the input data. The output arguments *MaxAmb* and *Amb* return the magnitude of the CAF plane’s peak and the matrix of complex values for the CAF plane. The output arguments *TauValues* and *FreqValues* are the ranges of TDOA and FDOA values computed over the CAF plane. This function also produces a CAF plane plot as seen in Figure 4-1.

The CAF_peak.m function uses the FFT method as described in Stein [2]. This method takes advantage of the fact that Equation 2-1 closely resembles the Fourier transform of the cross correlation of $s_1(t)$ and $s_2(t)$. In its discrete time form letting

$t = nT_s$ and $f = \frac{kf_s}{N}$, where T_s is the sample period, $f_s = \frac{1}{T_s}$ is the sampling frequency, *n* represents the individual sample numbers, and *N* is the total number of samples in the snapshot. Once these are inserted back into Equation 2-1, we get Equation 4-10:

$$CAF(\tau, k) = \sum_{n=o}^{N-1} [s_1(n) \cdot s_2^*(n-\tau)] e^{-j2\pi \frac{kn}{N}} \tag{4-10}$$

where s_1 and s_2 are the sampled signals in analytic format, τ is the time delay in samples, and $\frac{k}{N}$ is the frequency difference in digital frequency, or faction of the sample frequency. Note the similarity with the Discrete Fourier Transform (DFT) in Equation 4-11.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{kn}{N}} \tag{4-11}$$

Now replace *x(n)* with $[s_1(n)s_2^*(n-\tau)]$ and we get the discrete form of the CAF equation noted in Equation 4-10. This is the basis on which the function CAF_peak.m operates. While this function operates well for generating the CAF surface, the resolution is limited. The TDOA resolution is limited to 0.5 samples or 0.5T$_s$ seconds. The FDOA resolution is $\frac{0.5}{N}$ (digital frequency), or $\frac{0.5}{N} f_s$ Hertz. To improve the resolution enough to use in demonstrating the CAF-Map method the CAF surface was interpolated using a 2-D "cubic" interpolation in MATLAB®.

It should be noted that additional efficiencies could be taken advantage of by the realization that the correlation between *s₁* and *s₂* is computed efficiently using the FFT and the inverse FFT methods. This was demonstrated as a class project for SIGINT Systems I, "MATLAB Implementation of the Complex Ambiguity Function," by Hartwell and Jordan [13] at the Naval Postgraduate School.

C. MAPPING THE CAF SURFACE TO THE GROUND

Many methods were explored during this thesis to determine the most straightforward method of mapping the CAF plane to the surface of an area. This problem is very similar to the radio astronomy work in synthesis imaging explained in [14] and [15]. This effort is also similar to work that's been done in the field of Synthetic Aperture Radiometer (SAR) imaging [16], [17], and [18].

While it is easy to see how these methods are applicable to this problem, the method proposed by Mr. Al Buczek [1] was chosen due to its simplicity and ease of implementation. Some of these methods show promise and should be explored further, but are beyond the scope of this thesis. The function that implements the method put forth by Buczek is called the map_tdoa_fdoa.m function.

The function map_tdoa_fdoa.m, listed in Appendix A, is a function written in MATLAB® that maps the amplitude of the CAF surface to a 2-dimenstional x, y geographic map by matching the TDOA and FDOA values found in CAF surface to the TDOA and FDOA values in the lookup tables. These match the TDOA and FDOA values to an x, y coordinate. Once the coordinates are matched the amplitude information from each TDOA, FDOA pair in the CAF surface is mapped to the appropriate x and y coordinate to form the CAF-Map. The function is called from the main program called "caf_map.m" or it is invoked on the command line of the form:

[map,PtempX,PtempY]=map_tdoa_fdoa(tdoa_grid,fdoa_grid,Amb,dm,Fs,TauValues,FreqValues,Pe1,Pe2);

The input arguments *tdoa_grid* and *fdoa_grid* are the lookup tables for the TDOA and FDOA values computed in the function tdoa_fdoa_grid3D.m. The input argument *Amd* is the CAF surface produced by the CAF_peak.m function. The input argument *dm* is the desired x and y resolution of the CAF-Map image. The input argument *Fs* is the sampling rate in Hz of the receivers' digitizer. The input arguments *TauValues* and *FreqValues* are range of TDOA and FDOA values computed over the CAF plane and are the axes of the CAF surface. The input arguments *Pe1* and *Pe2* describe the area to calculate CAF-Map image. These arguments are two dimensional [x, y] in meters. The north-south direction is the y argument while the east-west is the x argument. It is assumed that the grid points are on the surface of a flat earth and that the altitude is zero meters. The output argument *map* is the complex map image of the CAF-Map. The output arguments *PtempX*, and *PtempY* are the x, and y, axes of the CAF-Map.

At the heart of the function is a simple algorithm where the TDOA and FDOA values are looked up for each x, y coordinate in TDOA and FDOA lookup tables and the amplitude from TDOA, FDOA coordinates of the CAF plane are mapped to the CAF-Map.

```
for x = 1:m
   for y = 1:n
      t = tdoa_grid(x,y);
      f = fdoa_grid(x,y);
      j = findnearest(TauValues,(t*Fs),0);
      i = findnearest(FreqValues,(f/Fs),0);
      map(x,y)=G(i,j);
   end
end
```

D. THE CAF-MAP SURFACE

The surface of each snapshot CAF-Map shows the TDOA and FDOA mapped to a flat earth. The structure of this surface is very similar to the one described in Dr. Michael Price's paper "Mathematics of Geolocation"[19]. Figure 4-7 shows a drawing of the surface described in Dr. Price's paper. The TDOA seems to modulate the surface of the FDOA surface, i.e. the amplitude of the FDOA surface depends on the TDOA. Figures 4-8 and 4-9 show a CAF-Map of a single snapshot. The CAF surface from which these maps were generated is shown in Figure 4-10.

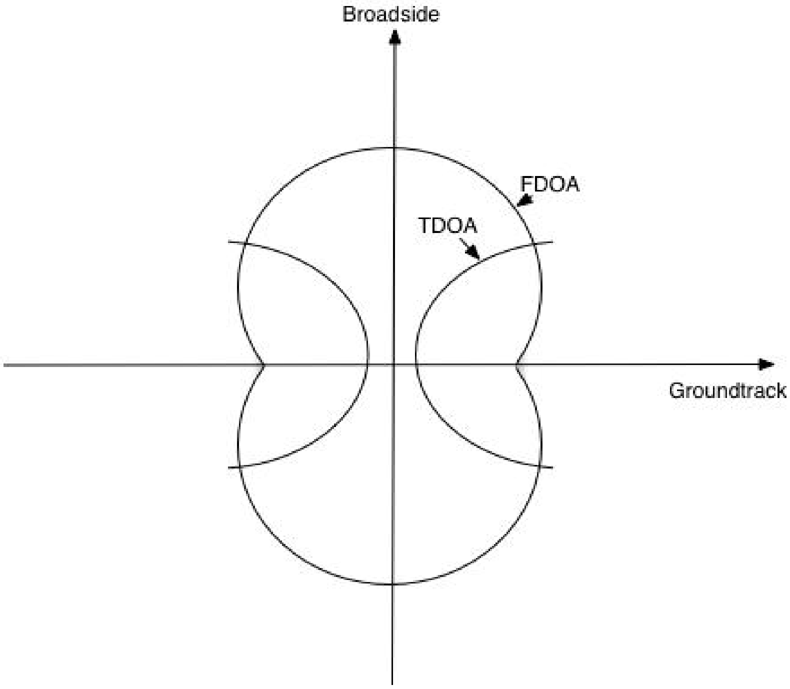


Figure 4-7: Surface explained by Price

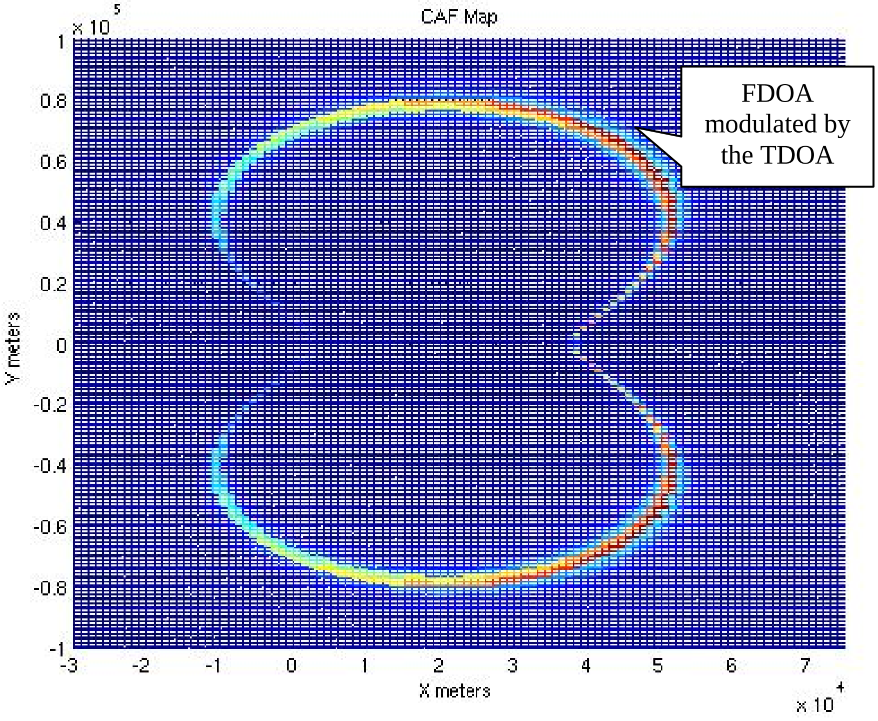


Figure 4-8: CAF-Map of a single snapshot

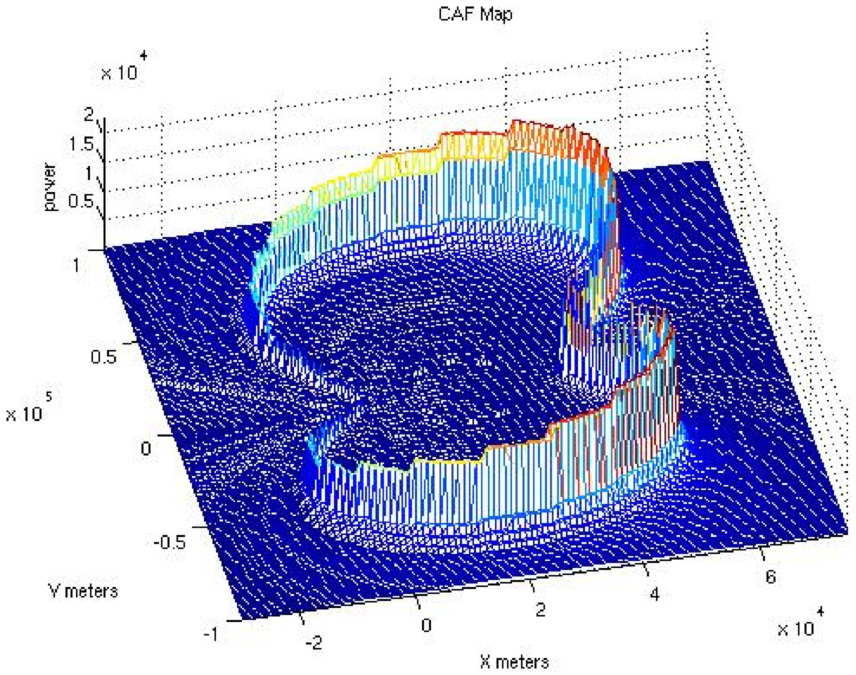


Figure 4-9: CAF-Map of a single snapshot

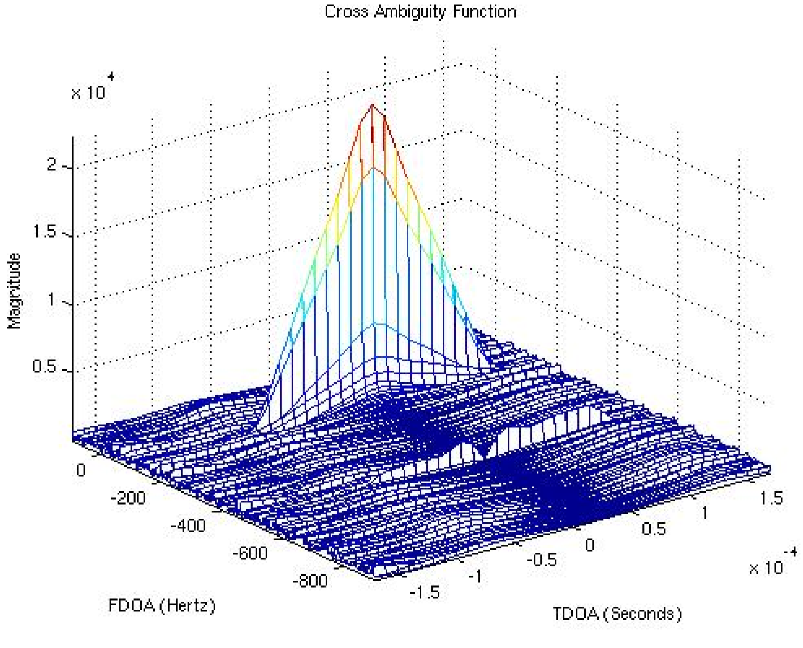


Figure 4-10: CAF surface used to generate Figures 4-8 and 4-9

E. COMBINING THE CAF-MAPS

A simple averaging method is used to combine several snapshot maps into the final map. Because the snapshots were collected in a non-coherent manner, the absolute values of the snapshot maps are averaged. It is also noted that the interpolation of the CAF plane may corrupt the phase information making the coherent combination of the maps non-optimal. When viewing each map of a sequence it is interesting to note how all of the surfaces have energy at the geolocation of the target in common on each map. Figures 4-12 through 4-16 show a sequence of snapshots illustrating how the energy of the surface rotates around the emitter's location. The Power Spectrum Density (PSD) of one of the collection channel's snapshots is shown is Figure 4-11. The collection pair was moving in the x direction, the lead followed by the trail separated by 20 km, traveling at 150 m/s, roughly 292 knots. Both of the platforms altitudes were 7.5 km or about 24,600 ft. The resolution of this example is 1 km.

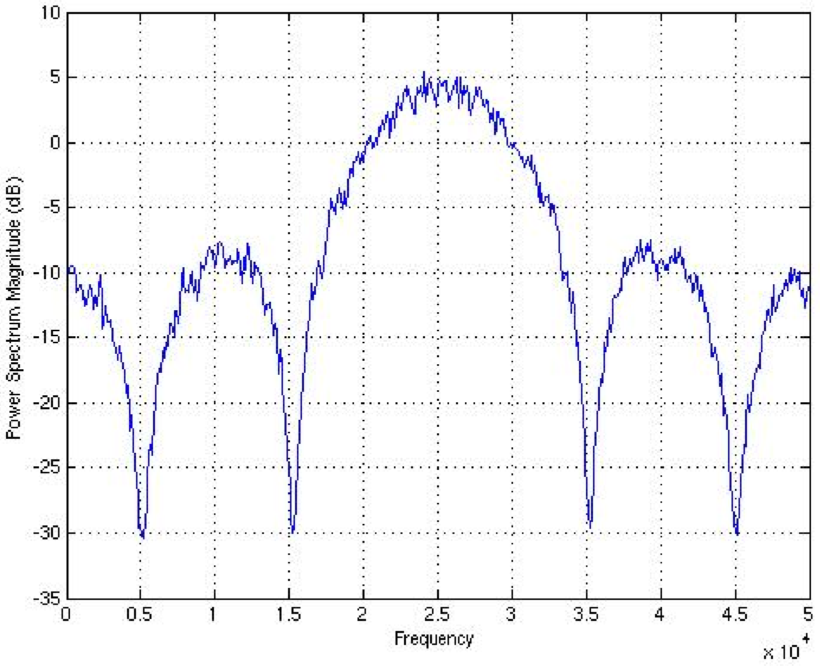


Figure 4-11: PSD of the collected signal

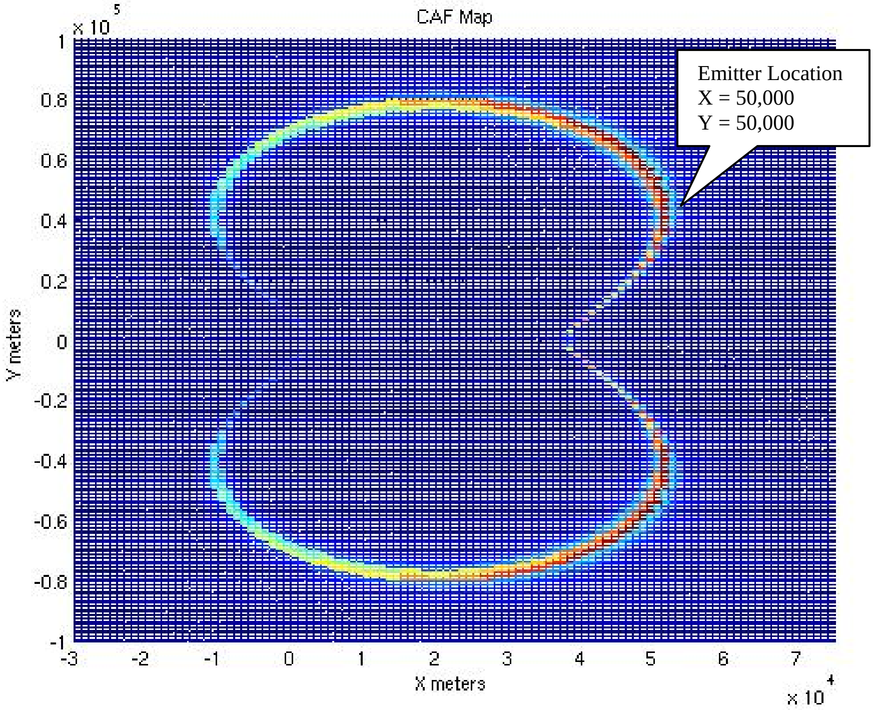


Figure 4-12: CAF-Map from collection pair at P1 = [10e3,0], P2 = [30e3,0] meters

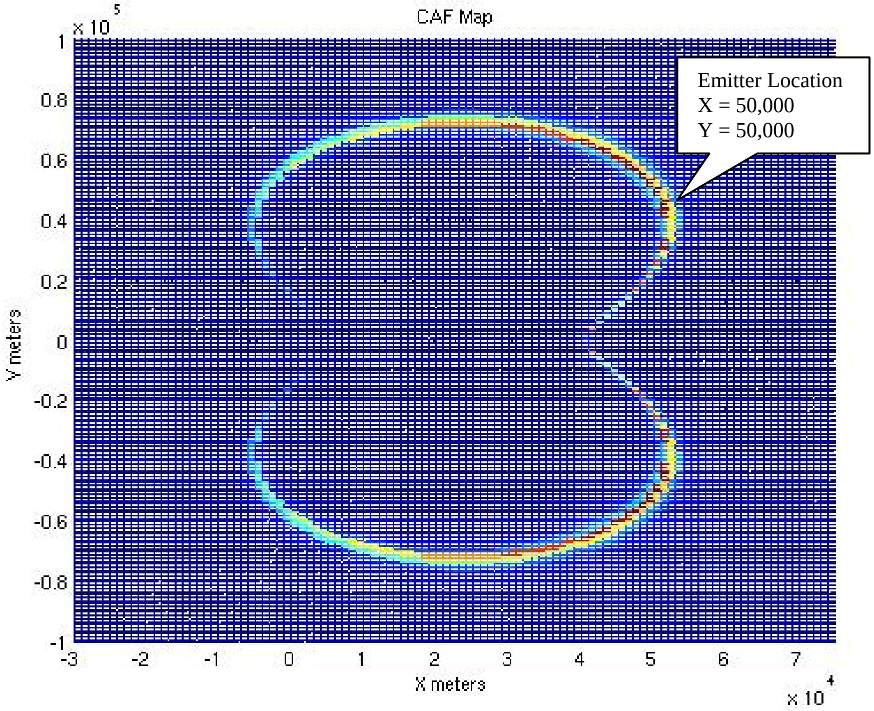
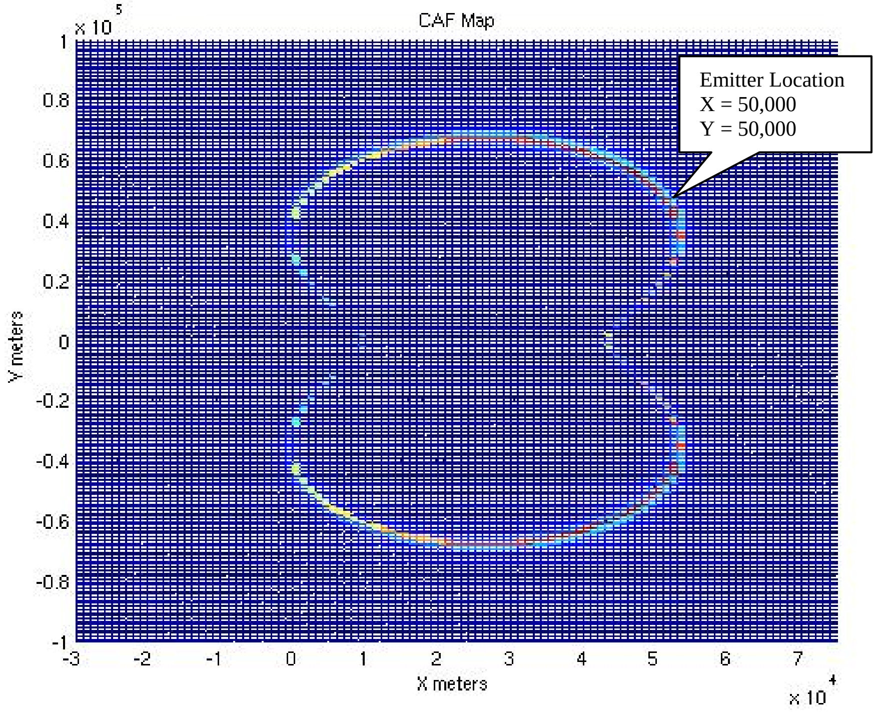
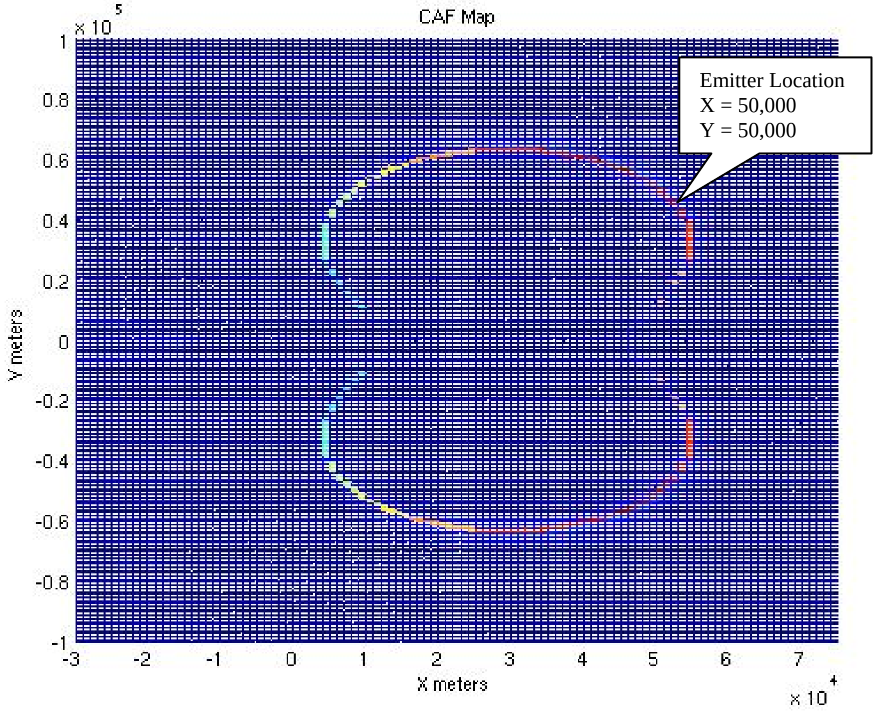


Figure 4-13: CAF-Map from collection pair at P1 = [13e3,0], P2 = [33e3,0] meters



Emitter Location
X = 50,000
Y = 50,000

Figure 4-14: CAF-Map from collection pair at P1 = [16e3,0], P2 = [36e3,0] meters



Emitter Location
X = 50,000
Y = 50,000

Figure 4-15: CAF-Map from collection pair at P1 = [19e3,0], P2 = [39e3,0] meters

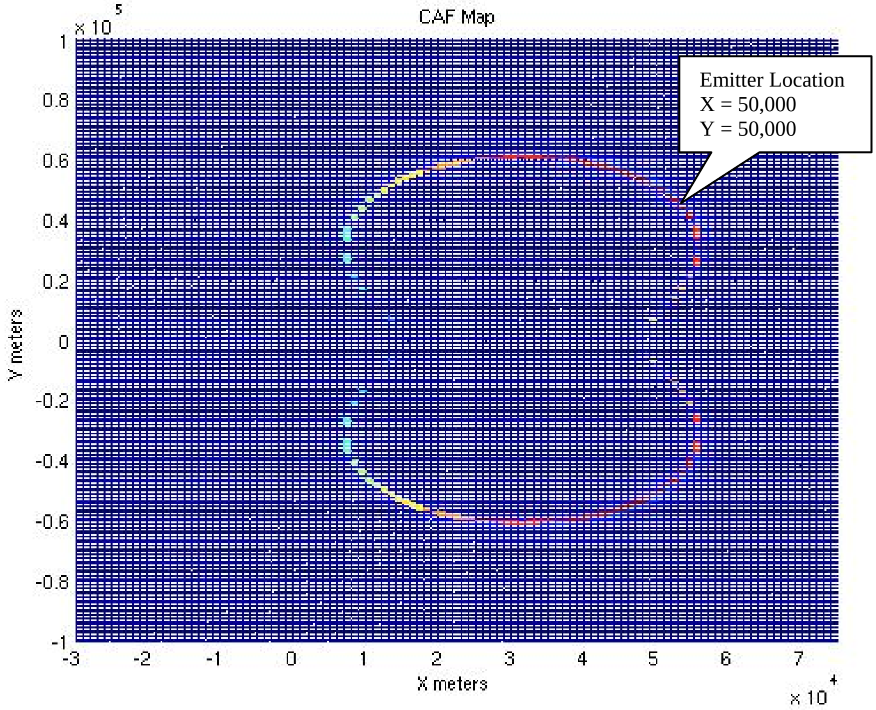


Figure 4-16: CAF-Map from collection pair at P1 = [21e3,0], P2 = [41e3,0] meters

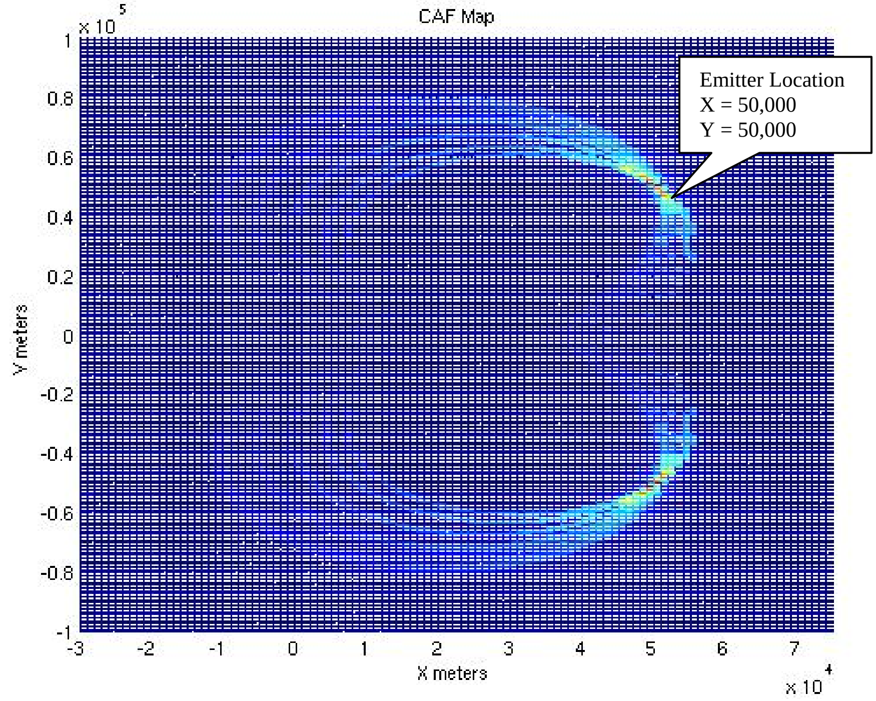


Figure 4-17: CAF-Map of the combined maps from Figures 4-11-4-15

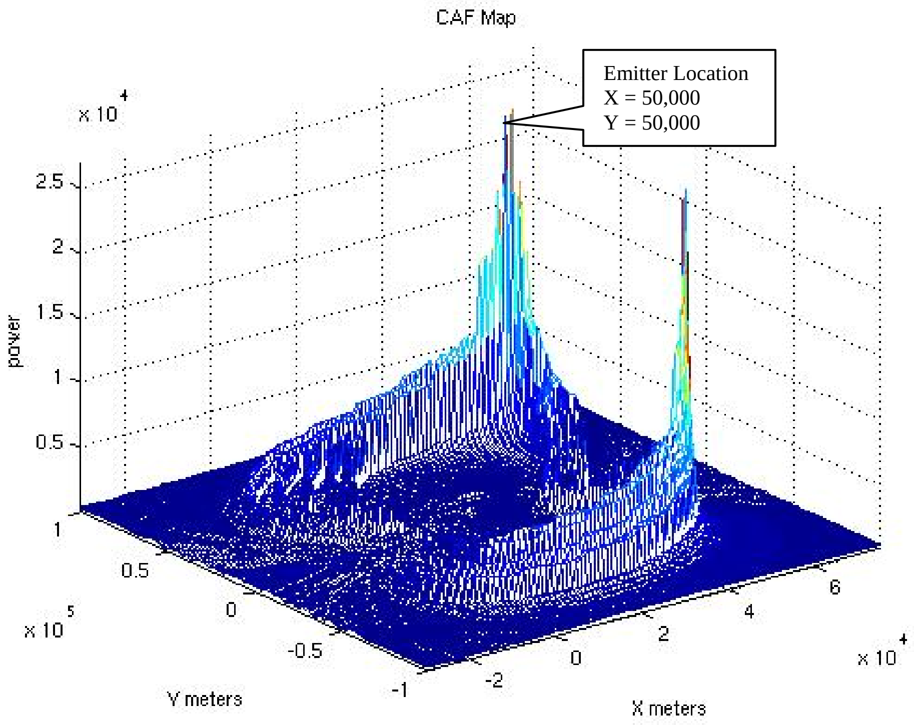


Figure 4-18: CAF-Map of the combined maps from Figures 4-12-4-16

From the combined CAF-Map shown in Figures 4-17 and 4-18, it is clear that the left-right ambiguity is still a problem. Peaks equal in magnitude were found at two locations, the first at the expected location of X = 50,000 & Y = 50,000 and at the mirrored location of X = 50,000 & Y = -50,000. The Collector's snapshot setup for this series of snapshots follows:

Carrier Frequency:	1000.025 MHz
Sampling Frequency:	100 kHz
Modulation Rate:	10 kbauds/sec
Modulation:	BPSK
Snap-duration:	32768 samples or 0.32768 seconds
Signal to Noise Ratio:	10 dB for each signal

Figures 4-19, 4-20, and 4-21 show the zoomed in peak of this CAF-Map. Note that the CAF interpolation was turned off for this series of Maps and the CAF-Map resolution was set to 1 km.

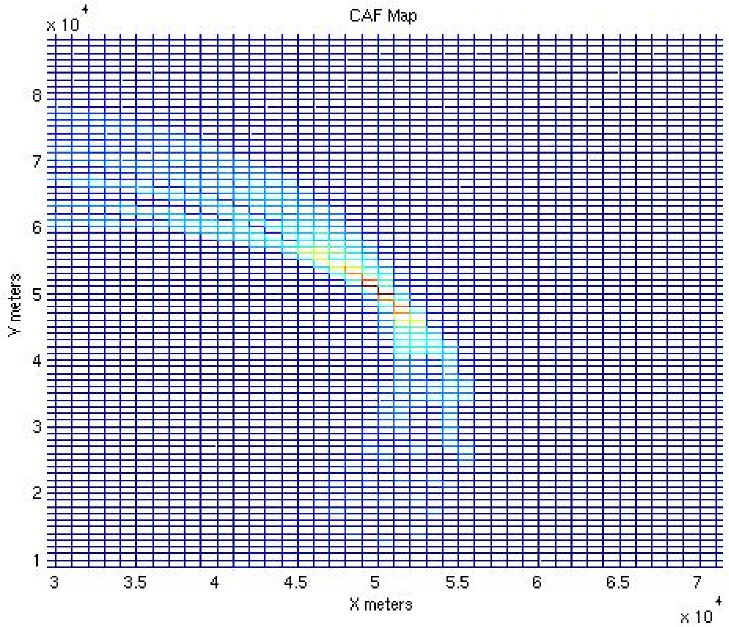


Figure 4-19: Zoomed X-Y plot of the peak at 50e3 by 50e3

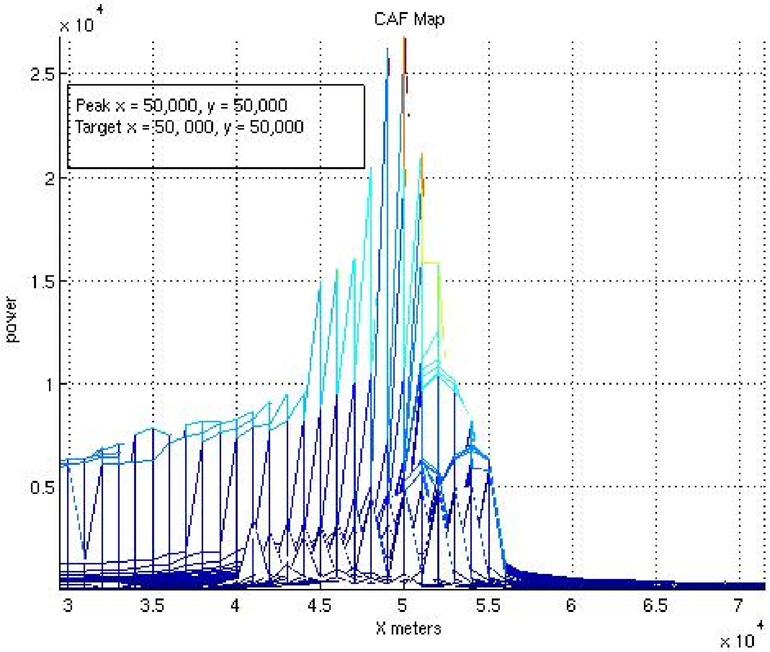


Figure 4-20: Zoomed X-Z plot of the peak at 50e3 by 50e3

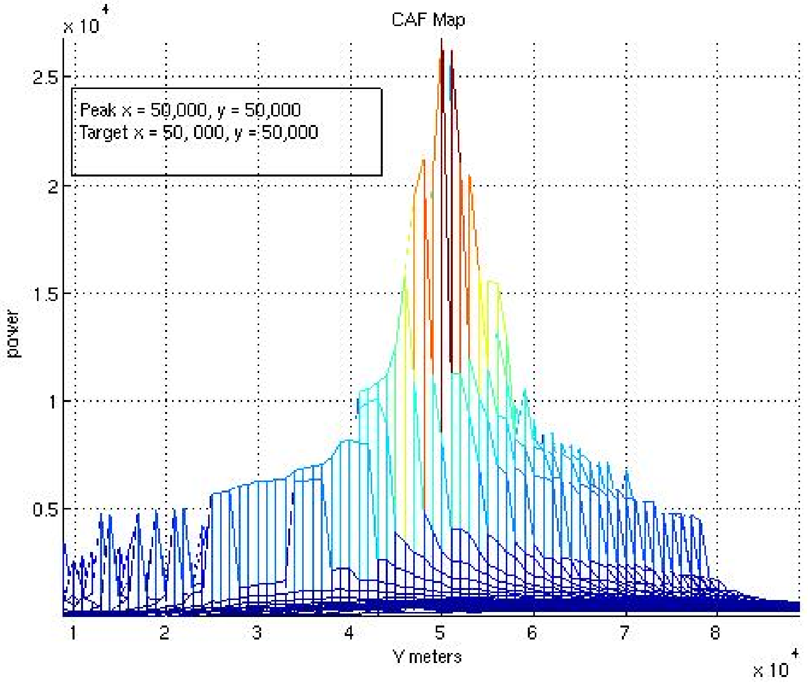


Figure 4-21: Zoomed Y-Z plot of the peak at 50e3 by 50e3

F. SIGNAL GENERATION

In order to test the CAF-Map method, signal pairs from known emitters with known TDOAs and FDOAs were required. To evaluate this method the program Sig_gen.m was used to generate known emitter signals with correct TDOA and FDOA based on the geometry of a pair of collection platforms. LCDR Joe J. Johnson wrote this program to test his implementation of a CAF tool in MATLAB [12]. This program was modified slightly to include the location of the collection platforms at the center of the snapshot.

This program Sig_gen.m, as listed in Appendix A, is a program written in MATLAB® that generates a pair of BPSK signals according to user-defined signal parameters and collector-emitter geometries as described in Chapter IV, Section A. The function is invoked in the command line of the form:

[Sa1,Sa2,S1,S2,Pcc1,Pcc2] = sig_gen;

This program has no input arguments since the user is queried for all required parameters. Four of the six output arguments are returned as signal vectors. *Sa1* and *Sa2* are the two generated signals in analytic format. *S1* and *S2* are the real valued signals generated.

Both sets of signal vectors are time domain vectors. The two output arguments *Pcc1* and *Pcc2* describe each of the collector's positions in the middle of the collection snapshot. The format of the position vectors are each three dimensional entries and are in [x, y, z] form in meters. The north-south direction is the y argument with east-west being the x argument. The collector's altitude is in the z direction.

The user is asked a series of questions to gather the information to generate the signals. The user is first asked to input the position and velocity vector information of the two collectors at "time 0." All position and velocity information are entered in [x, y, z] form. The north-south direction is the y argument with east-west being the x argument. The collector's altitude is in the z direction. All three arguments are in meters. The velocity vectors are each three dimensional entries and are in [Vx, Vy, Vz] form. These describe each collector's velocity vector and are in meters/seconds. With the geometry entries complete the user is asked for information on the collected signal. The user is asked for the carrier frequency and sample rate both in Hz. Note that the program will alias the carrier frequency so the user must choose a frequency that will alias nicely into the Nyquest bandwidth. The user is then asked for the symbol rate of the BPSK signal. Again the user must be careful to choose signals whose bandwidth remains within the Nyquist bandwidth. The user is then asked for the numbers of samples in the snapshot and then is asked for each signal's Signal-to-Noise Ratio SNR. After this entry the program will generate the output arguments. The program will also print on the screen a TDOA and FDOA values for the beginning and end of the snapshot.

Chapter V shows several examples of the CAF-Map method in uses. It also shows the importance of the collector geometry and separation in emitter location and in eliminating the left-right ambiguity problem. The last two examples demonstrate that this method works well by geolocating several co-channel emitters.

V. EXAMPLES

Several simulations with different geometry were used to demonstrate the effects of collector separation and geometry on the CAF-Map results. Additional simulations were generated to include a co-channel interfering signal.

A. SCENARIO #1

In the first simulation, there is an emitter at location x = 10 km y = 10 km. The collection platforms were separated by 2 km and are flying in a lead trail configuration. Figure 5-1 shows the geometry of this scenario. The CAF-Map resolution for this scenario was set to 100 meters.

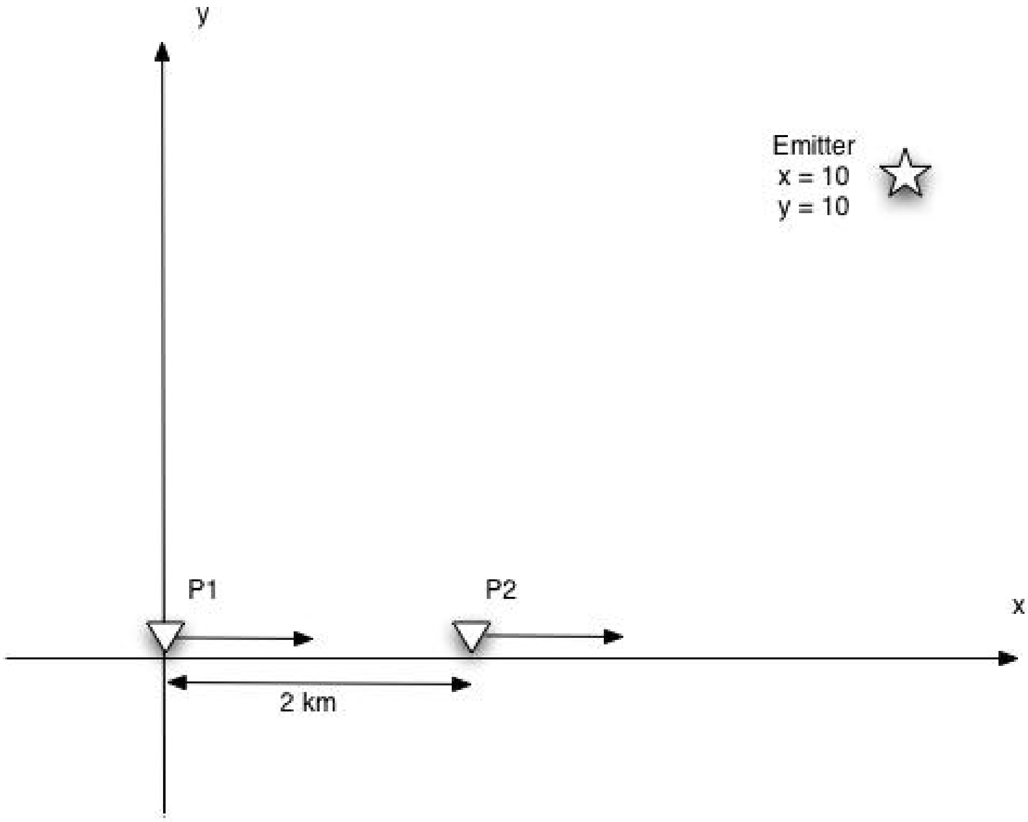


Figure 5-1: Collector Geometry for Scenario 1

The platforms were at an altitude of 7.5 km and are flying at 100 m/s. A snapshot was taken every twenty seconds. The collectors moved 2 km between each snapshot along the x-axis. The Collector's snapshot setup for this series of snapshots follows:

Carrier Frequency:	1000.025 MHz
Sampling Frequency:	100 kHz
Modulation Rate:	10 kbauds/sec
Modulation:	BPSK
Snap-duration:	32768 samples or 0.32768 seconds
Signal to Noise Ratio:	10 dB for each signal
Duration between snapshots:	20 seconds

Each CAF-Map illustrated will have the collector's position in x and y coordinates in the title of the figures.

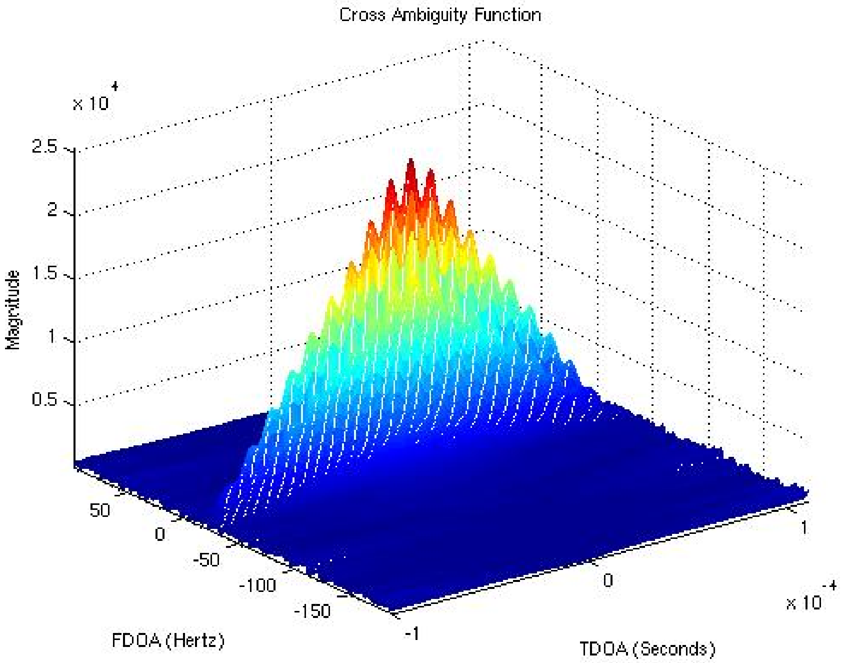


Figure 5-2: CAF surface with the collectors at P1 = [0,0], P2 = [2e3,0]

Illustrated in Figure 5-2 is the CAF of the first snapshot. This surface shows a good correlation in both time and frequency and should produce good geolocation results. Figures 5-3 through 5-12 show individual CAF-Maps for each snapshot. Again note how the surfaces revolve around the emitter location at 10 km and 10 km. These figures also show the left-right ambiguity problem.

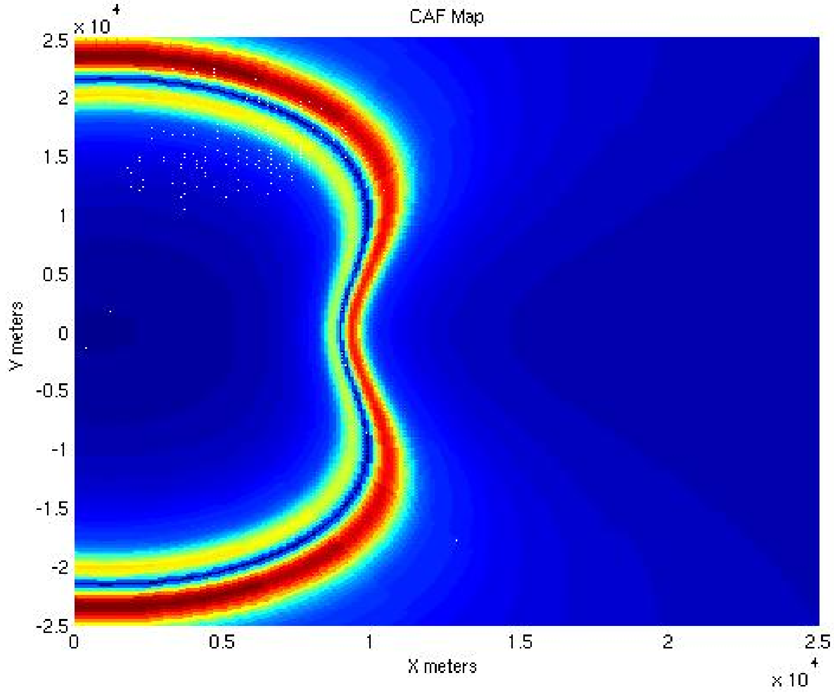


Figure 5-3:   CAF-Map from collection pair at P1 = [0,0], P2 = [2e3,0] meters

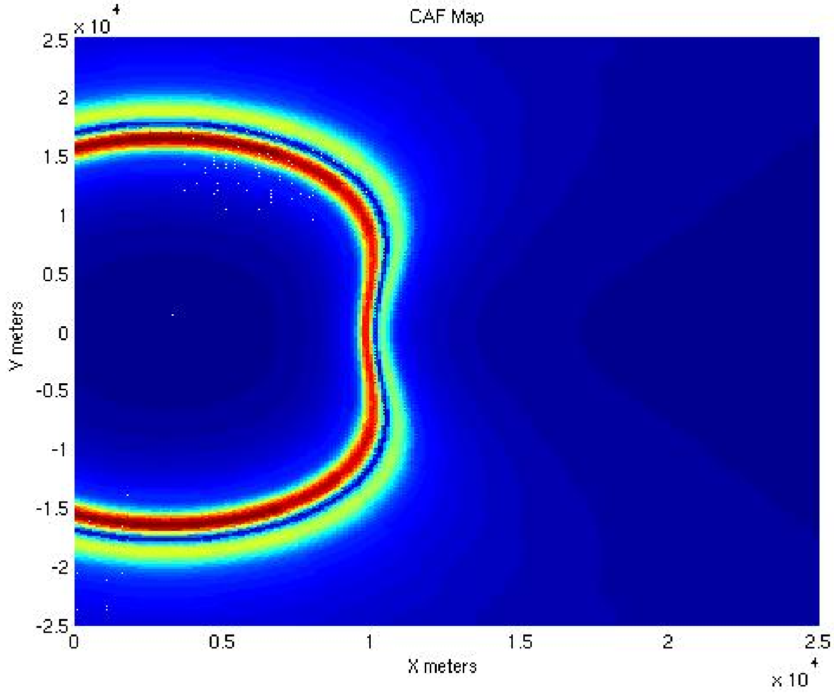


Figure 5-4:   CAF-Map from collection pair at P1 = [2e3,0], P2 = [4e3,0] meters

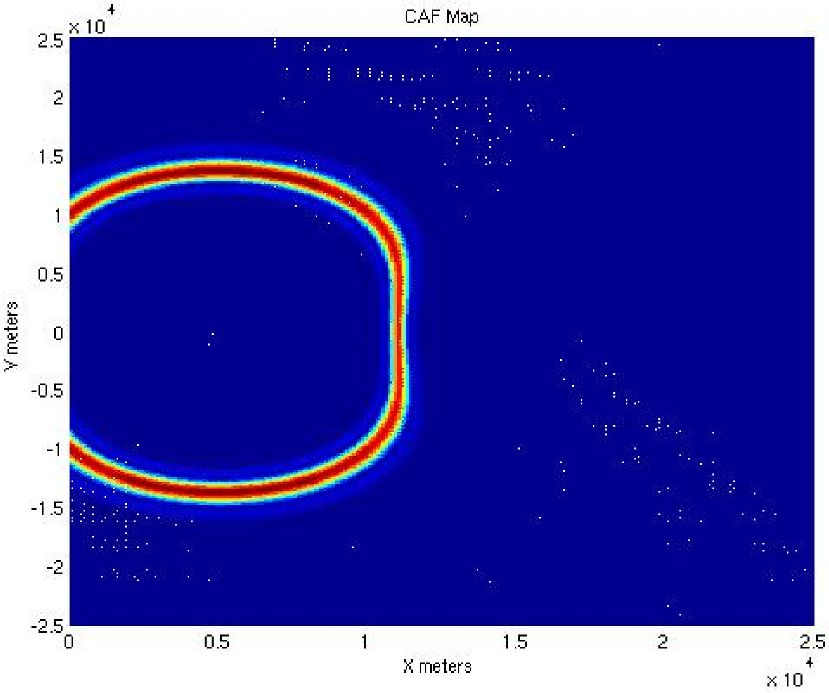


Figure 5-5: CAF-Map from collection pair at P1 = [4e3,0], P2 = [6e3,0] meters

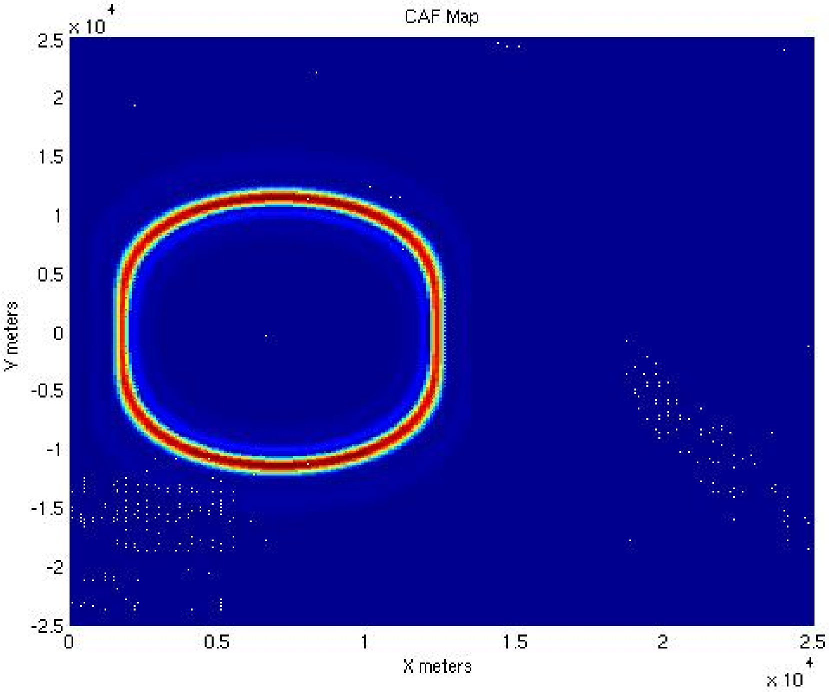


Figure 5-6: CAF-Map from collection pair at P1 = [6e3,0], P2 = [8e3,0] meters

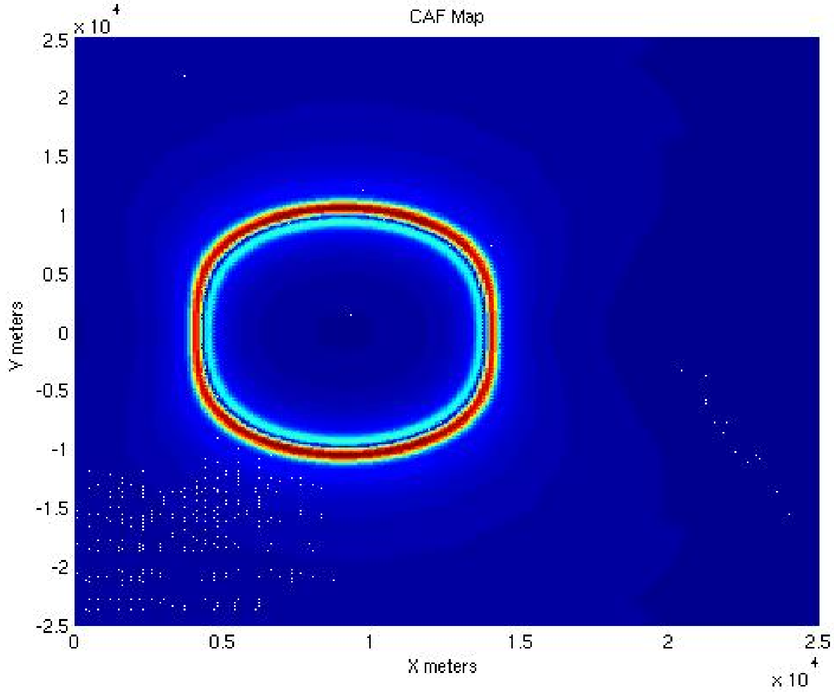


Figure 5-7: **CAF-Map from collection pair at P1 = [8e3,0], P2 = [10e3,0] meters**

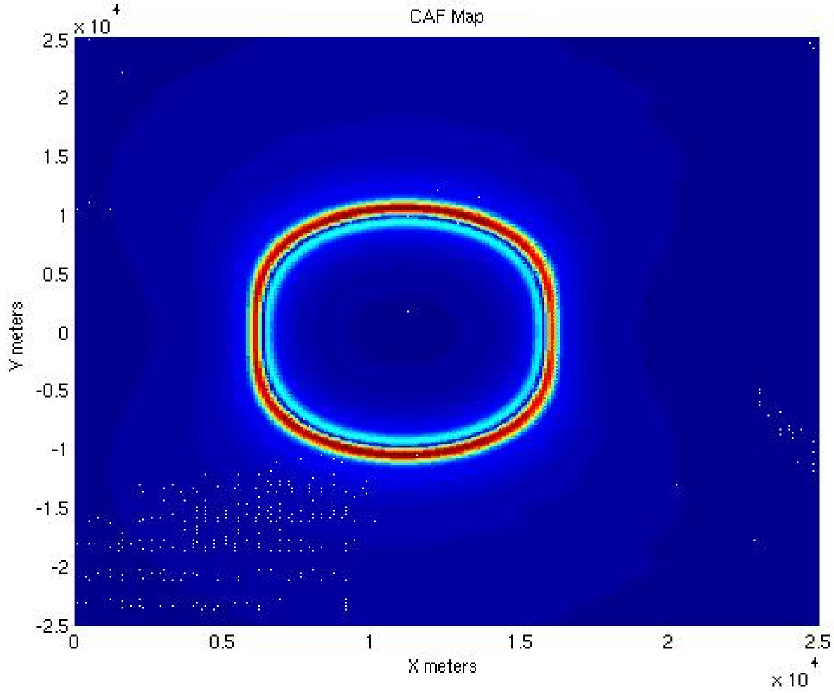


Figure 5-8: **CAF-Map from collection pair at P1 = [10e3,0], P2 = [12e3,0] meters**

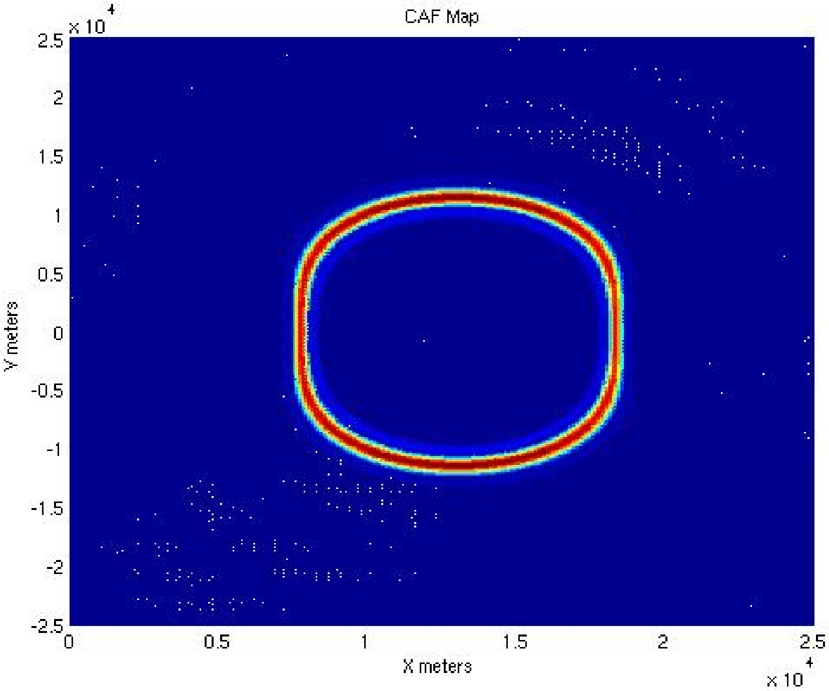


Figure 5-9: CAF-Map from collection pair at P1 = [12e3,0], P2 = [14e3,0] meters

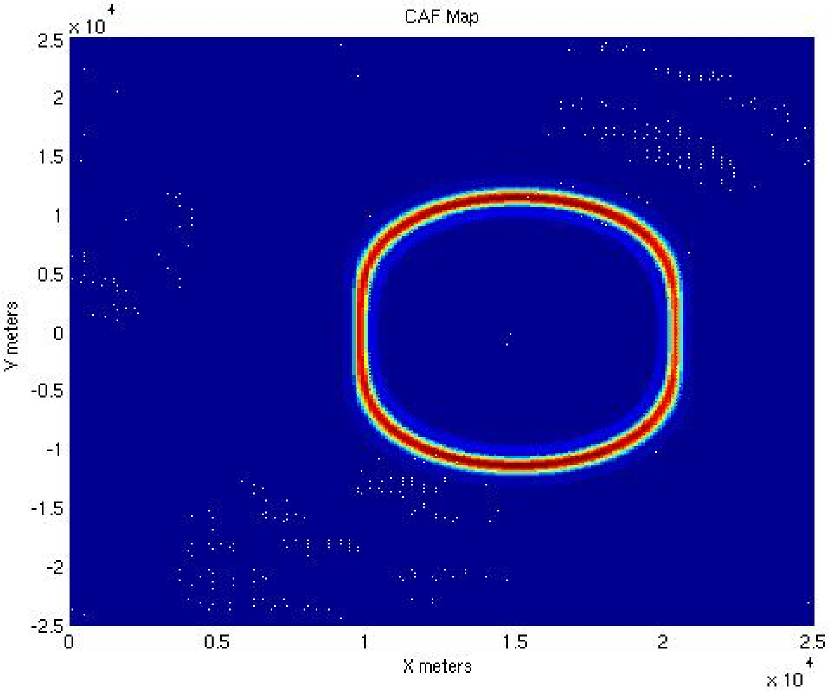


Figure 5-10: CAF-Map from collection pair at P1 = [14e3,0], P2 = [16e3,0] meters

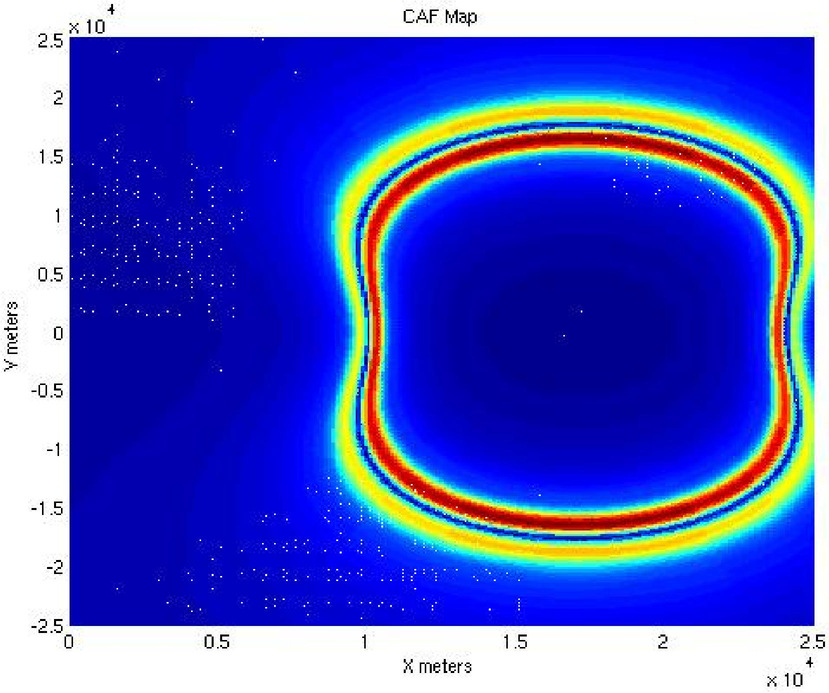


Figure 5-11: CAF-Map from collection pair at P1 = [16e3,0], P2 = [18e3,0] meters

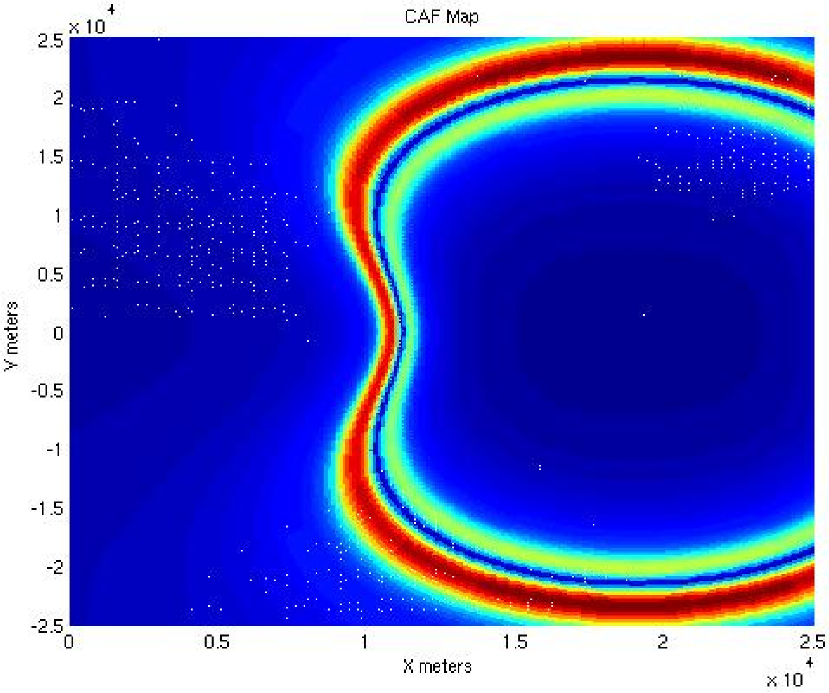


Figure 5-12: CAF-Map from collection pair at P1 = [18e3,0], P2 = [20e3,0] meters

The combined CAF-Map is shown in Figures 5-13 and 5-14. While not providing the peak in the correct location, the true location is within the top 10 % of the energy of the peak.

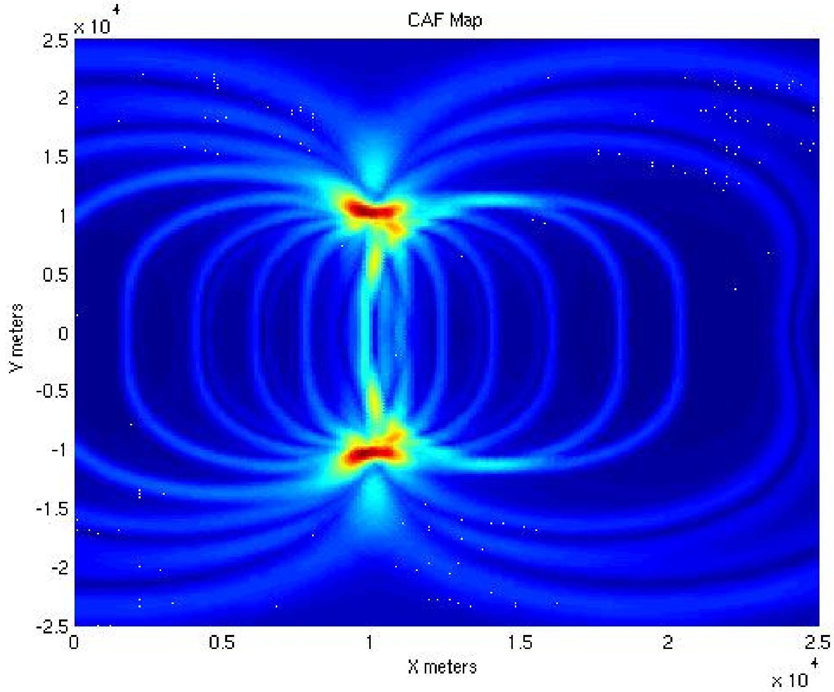


Figure 5-13: X-Y CAF-Map of the combined Maps

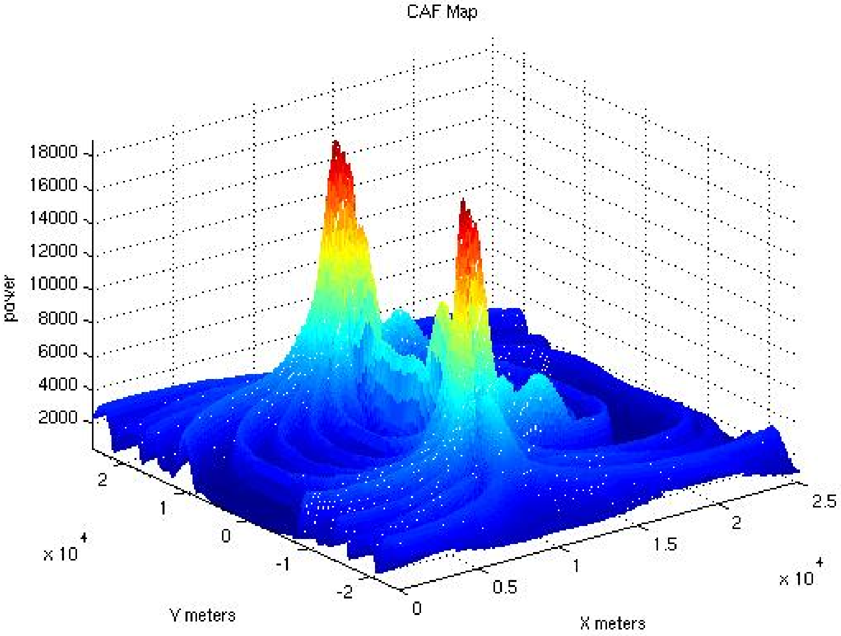


Figure 5-14: CAF-Map of the combined Maps

As noted before, the CAF map still has a left-right ambiguity problem in using this geometry. Figures 5-15 and 5-16 show details of the peak detected in the combine CAF-Map.

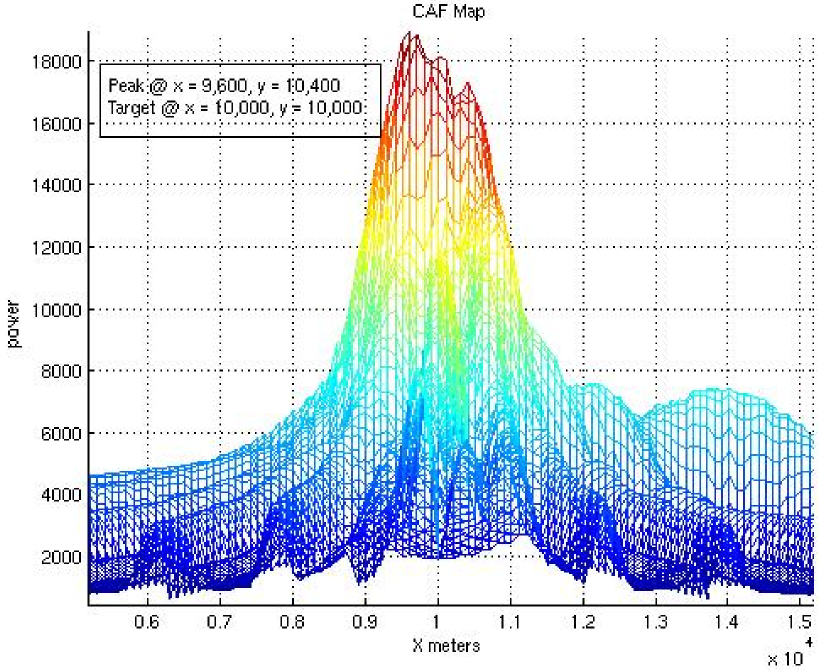


Figure 5-15: X-Z CAF-Map

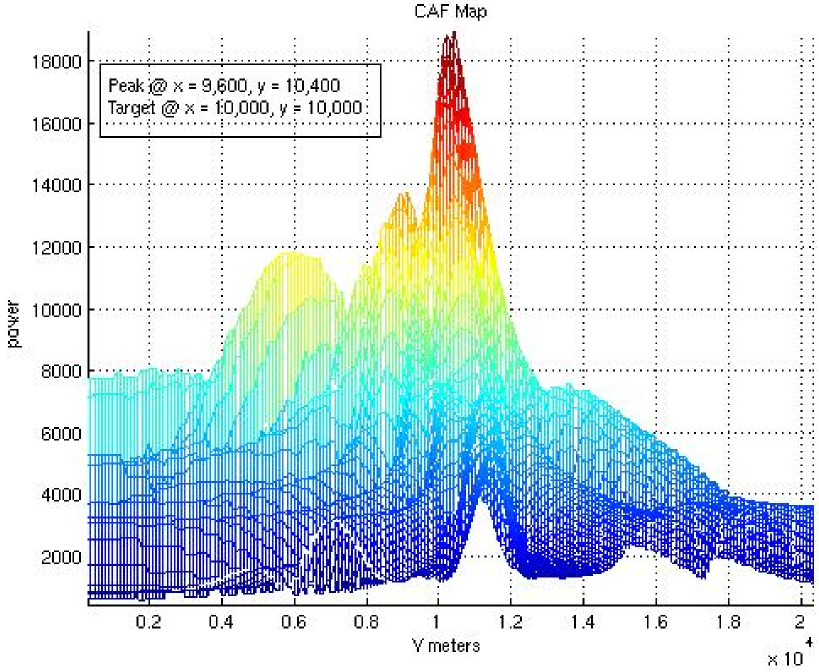


Figure 5-16: Y-Z CAF-Map

The result of scenario 1 showed an encouraging miss distance of only 565.7 meters. Again the CAF-Map resolution was set to 100 meters for this scenario.

B. SCENARIO #2

As in the first simulation the emitter is located at x = 10 km y = 10 km. However, this time to combat the left-right ambiguity problem the collection platforms were separated by 2.8284 km and instead of flying in a lead trail configuration collector 2 is offset in the y direction by 2 km. The CAF-Map resolution for this scenario was set to 100 meters. Figure 5-17 shows the geometry of this scenario.

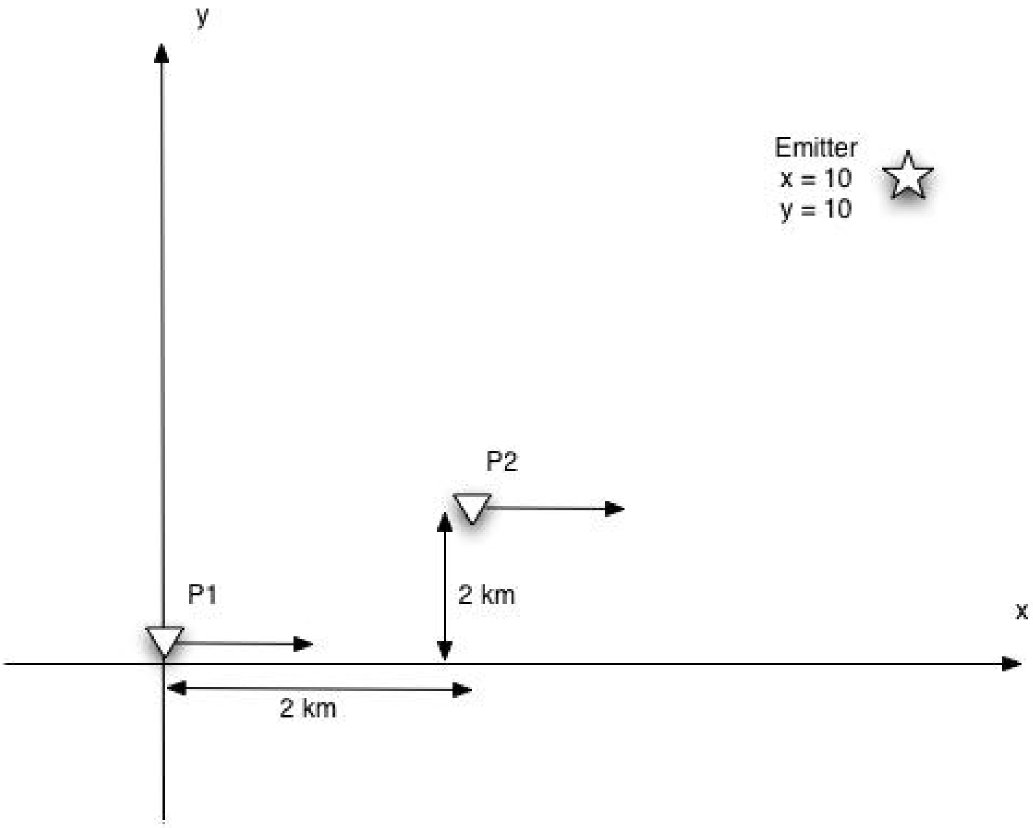


Figure 5-17: Collector geometry for Scenario 2

As before, the platforms were at an altitude of 7.5 km and are flying at 100 m/s. A snapshot was taken every twenty seconds. The collectors moved 2 km between each snapshot along the x-axis. The Collector's snapshot setup for this series of snapshots follows:

Carrier Frequency:	1000.025 MHz
Sampling Frequency:	100 kHz
Modulation Rate:	10 kbauds/sec
Modulation:	BPSK
Snap-duration:	32768 samples or 0.32768 seconds
Signal to Noise Ratio:	10 dB for each signal
Duration between snapshots:	20 seconds

Figures 5-18 through 5-27 show individual CAF-Maps for each snapshot of Scenario 2. Again note how the surfaces revolve around the emitter location at 10 km and 10 km.

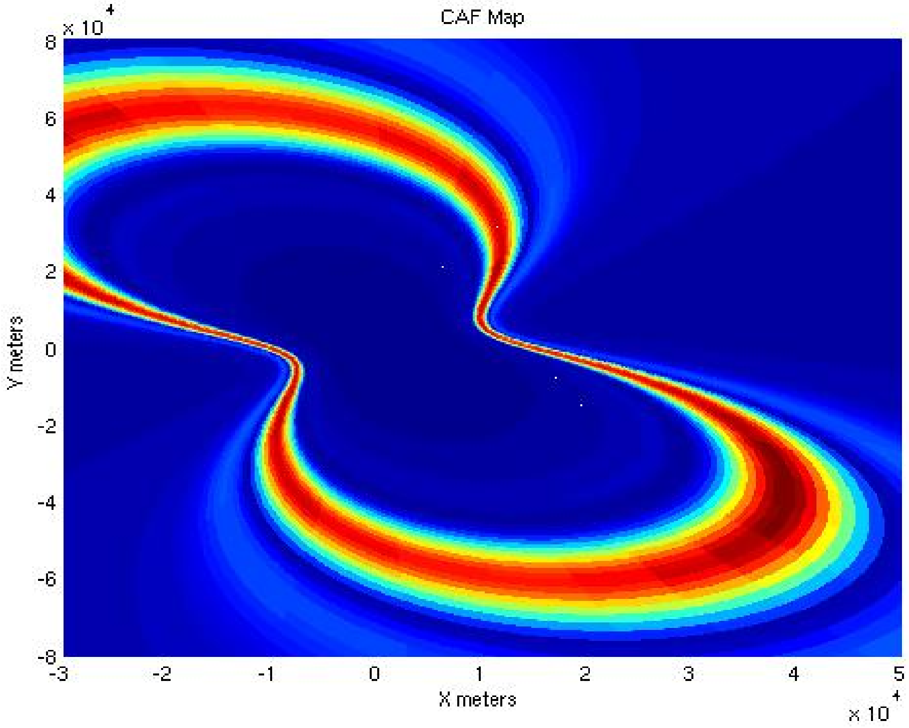


Figure 5-18: CAF-Map from collection pair at P1 = [0,0], P2 = [2e3,2e3] meters

In Figure 5-18 note the lack of symmetry along the ground track (x-direction). By offsetting the collection platforms in the ground track the FDOA surface has rotated to match the geometry. This is important to note because the TDOA is not affected by this change as discussed in Section II, B, 4. In the remainder of the CAF-Maps for this scenario note the rotation about the emitter.

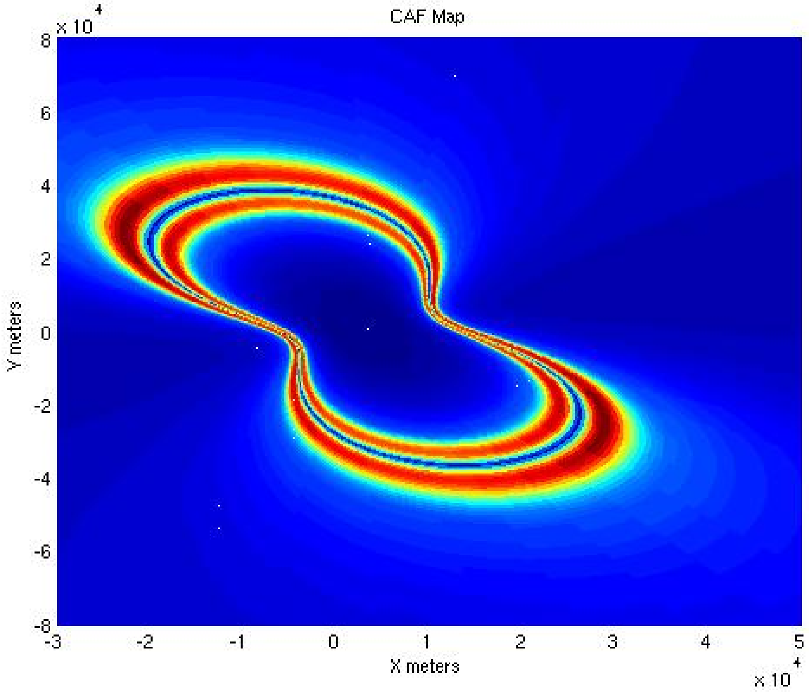


Figure 5-19: CAF-Map from collection pair at P1 = [2e3,0], P2 = [4e3,2e3] meters

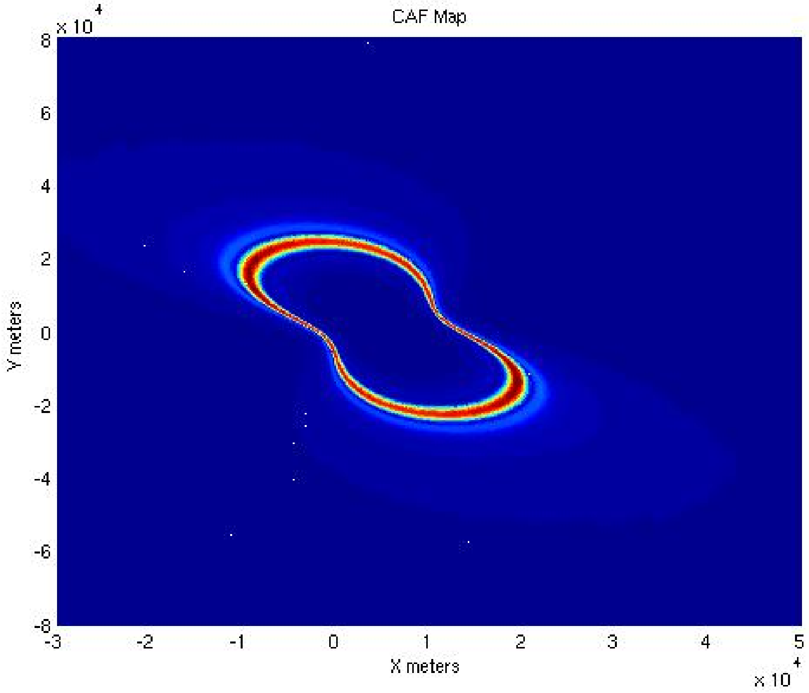


Figure 5-20: CAF-Map from collection pair at P1 = [4e3,0], P2 = [6e3,2e3] meters

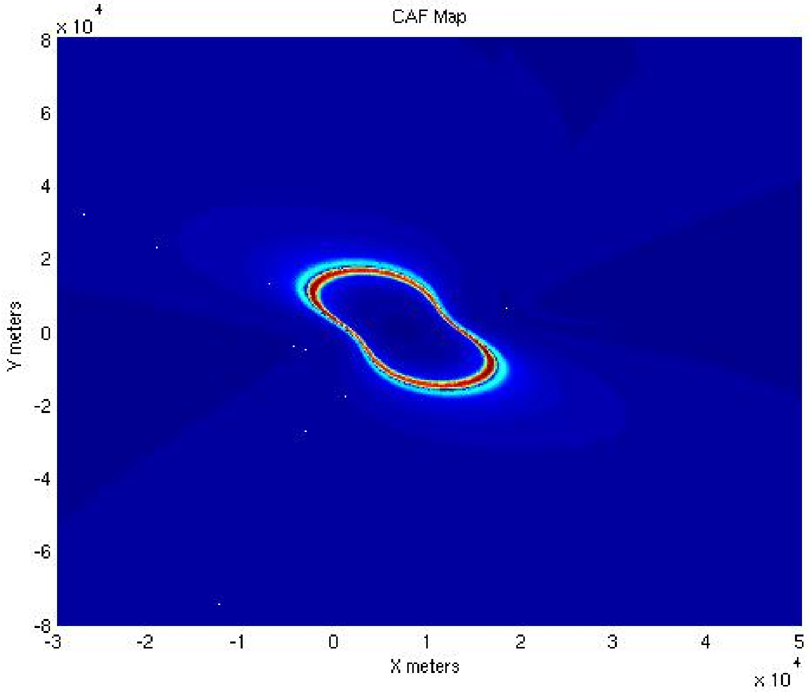


Figure 5-21: CAF-Map from collection pair at P1 = [6e3,0], P2 = [8e3,2e3] meters

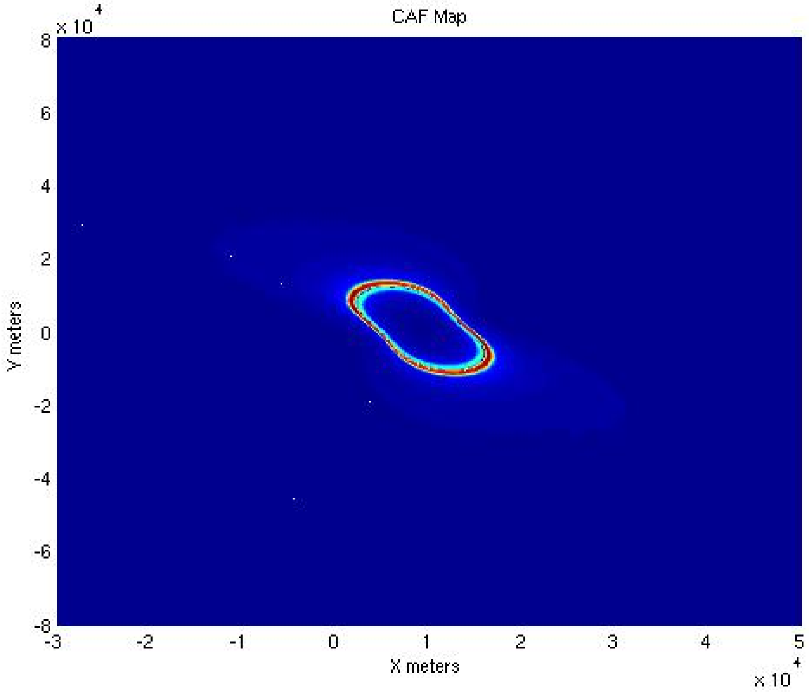


Figure 5-22: CAF-Map from collection pair at P1 = [8e3,0], P2 = [10e3,2e3] meters

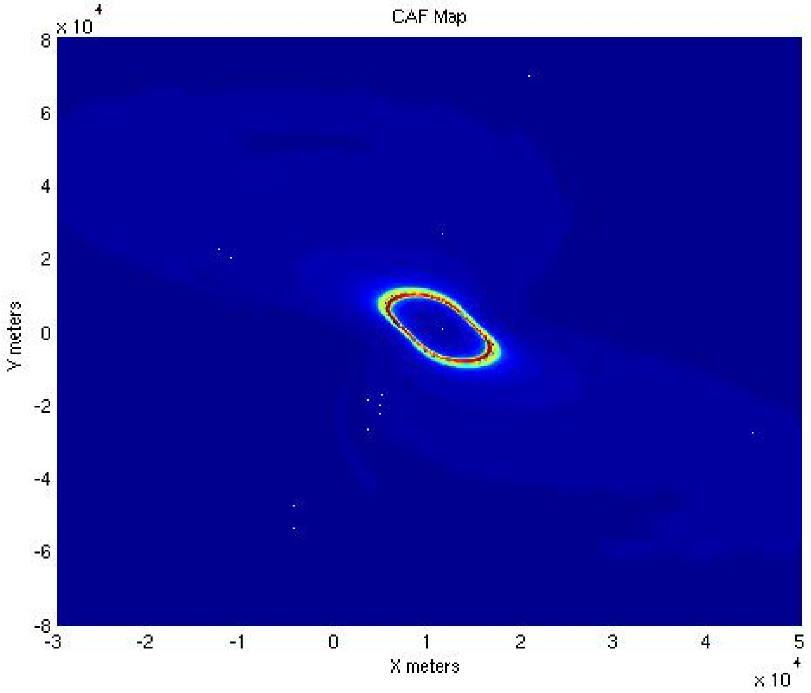


Figure 5-23: CAF-Map from collection pair at P1 = [10e3,0], P2 = [12e3,2e3] meters

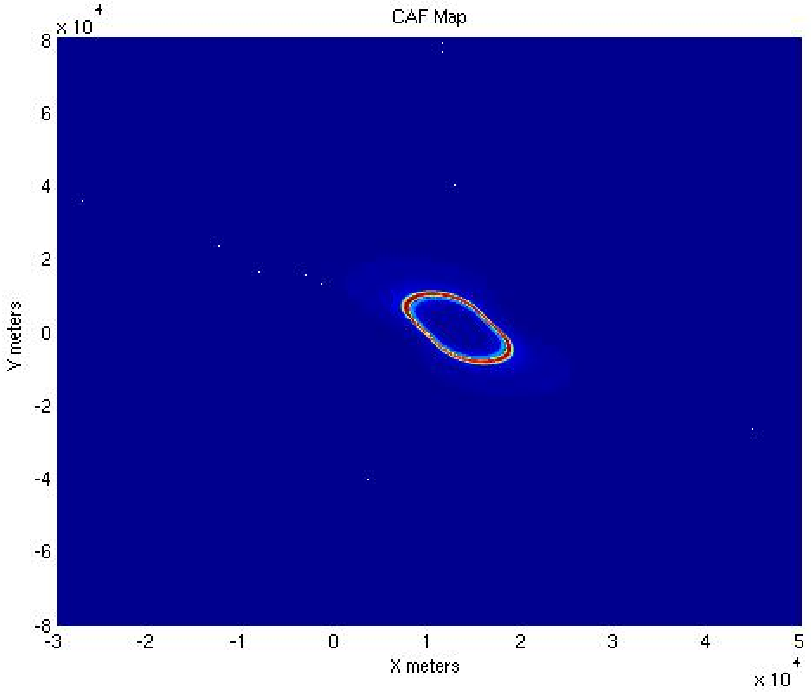


Figure 5-24: CAF-Map from collection pair at P1 = [12e3,0], P2 = [14e3,2e3] meters

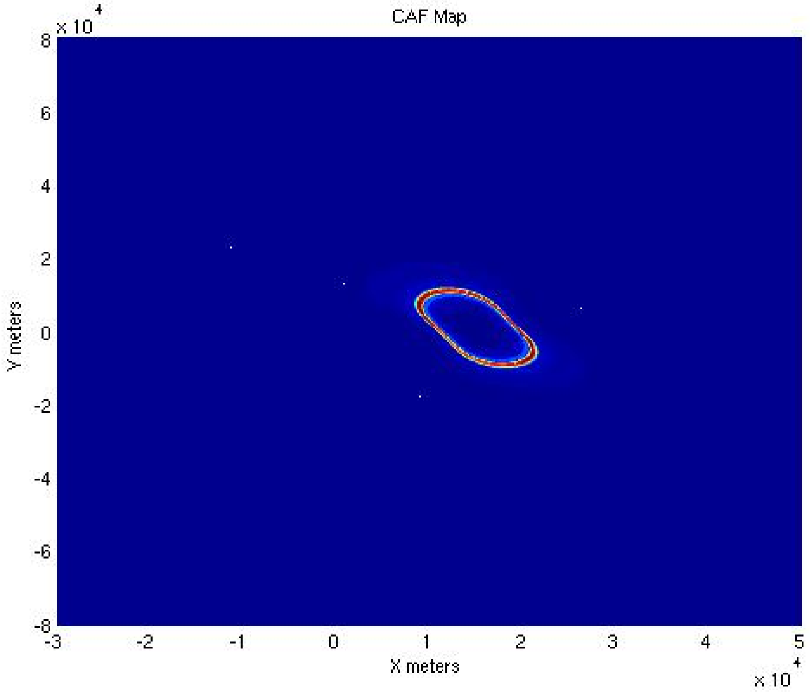


Figure 5-25: CAF-Map from collection pair at P1 = [14e3,0], P2 = [16e3,2e3] meters

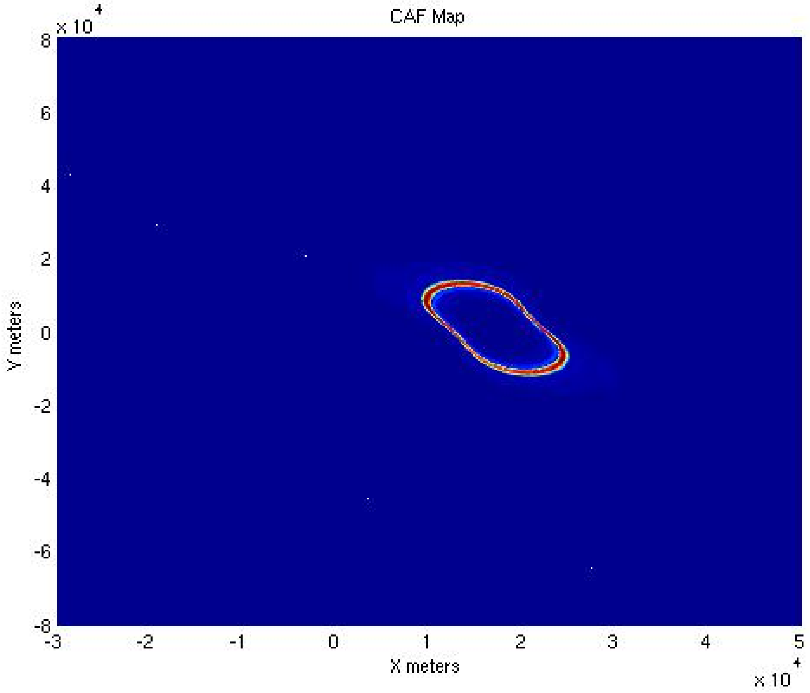


Figure 5-26: CAF-Map from collection pair at P1 = [16e3,0], P2 = [18e3,2e3] meters

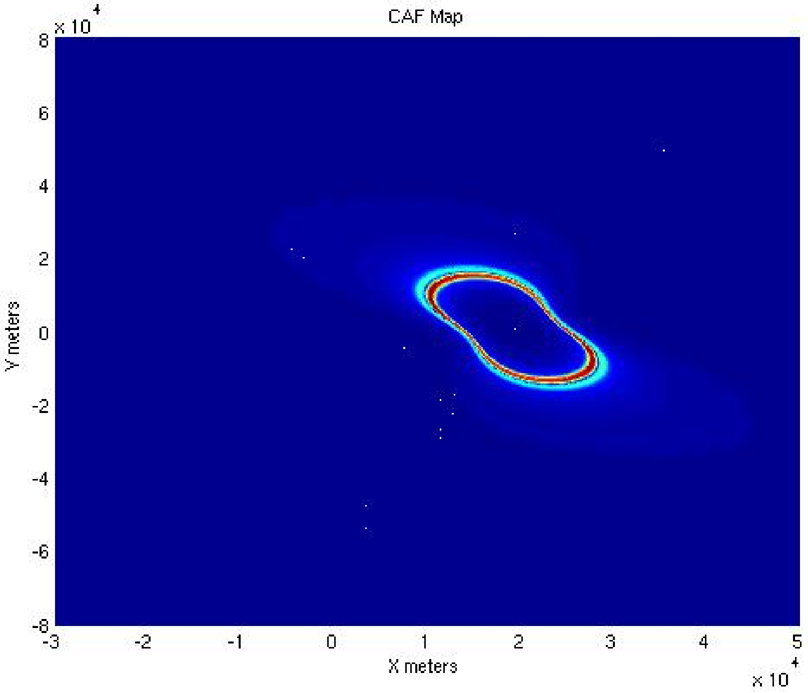


Figure 5-27: CAF-Map from collection pair at P1 = [18e3,0], P2 = [20e3,2e3] meters

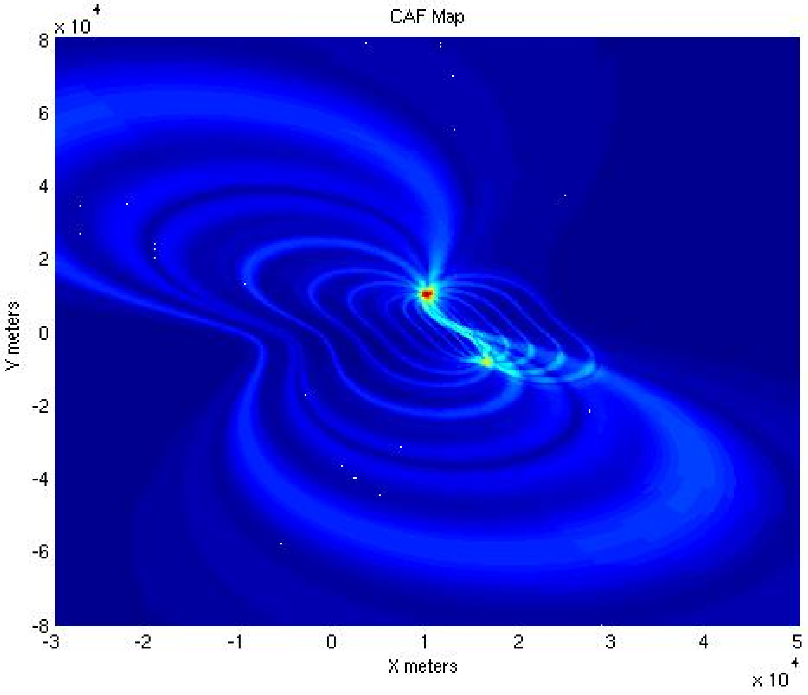


Figure 5-28: X-Y CAF-Map of combined Maps

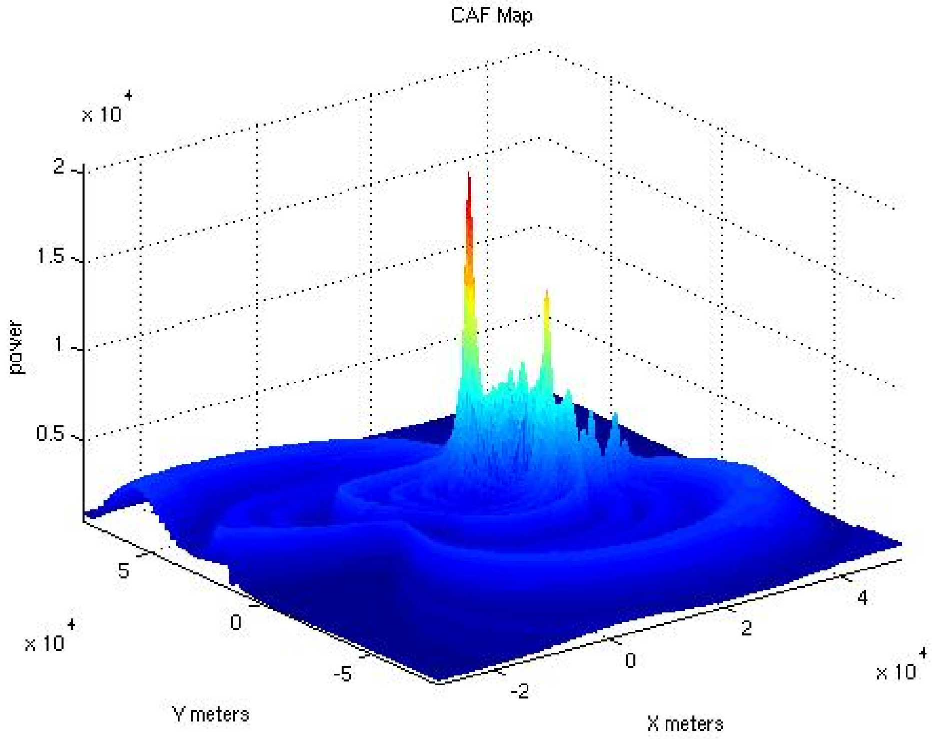


Figure 5-29: Combined CAF-Map surface

In Figures 5-28 and 5-29 the rotation around the emitter is clearly seen. Figures 5-30 and 5-31 show the details about the peak detected in the CAF-Map. In this scenario, the miss distance was only 141.4 meters and there were no left-right ambiguity problems. This showed very encouraging results considering the resolution for the CAF-Map was set to 100 meters. This scenario showed the effects of small changes in the geometry of the collection platforms.

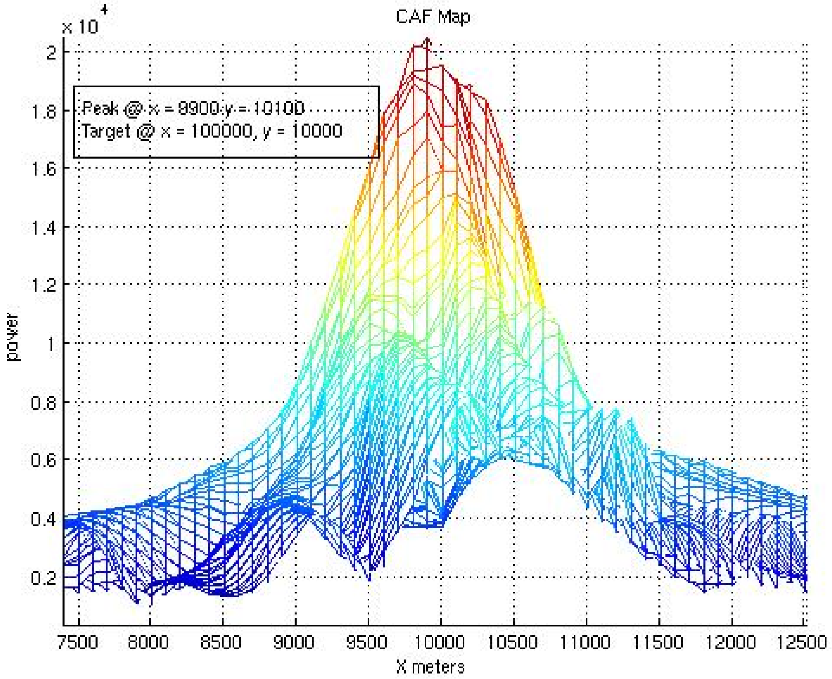


Figure 5-30: X-Z CAF-Map

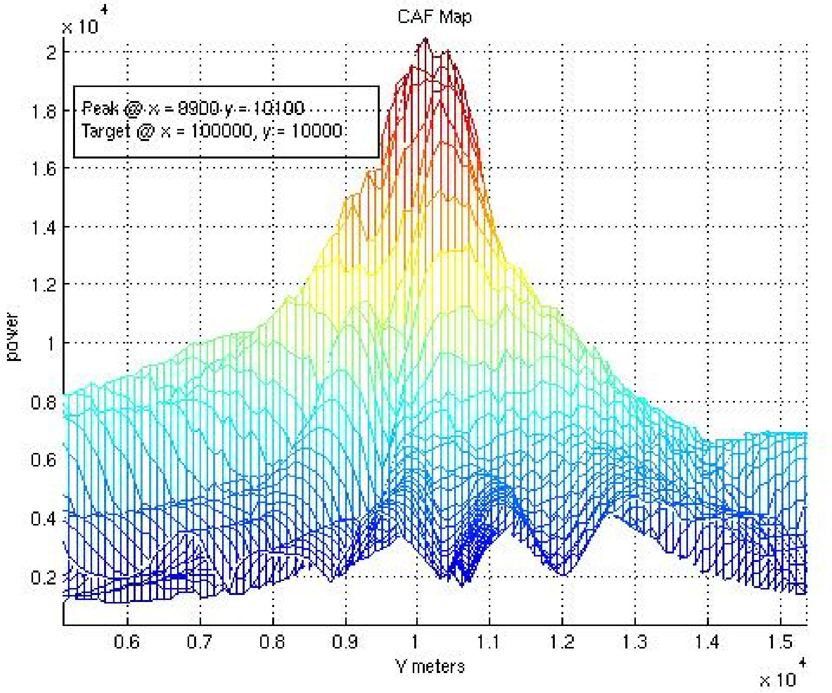


Figure 5-31: Y-Z CAF-Map

C. SCENARIO #3

This scenario was developed to demonstrate the co-channel capability of the CAF-Map method. In this scenario, two emitters were located in the mapped area. The first emitter was located at x = 30 km, y = 70 km, and the second emitter was located at x = 50 km, y = 50 km. As in Scenario 2, to combat the left-right ambiguity problem the collection platforms were separated along the cross-track by 5 km as well as along the ground-track by 10 km. The straight line separation was 11,180 meters. The CAF-Map resolution for this scenario was set to 1000 meters and the CAF interpolation was turned off. Figure 5-32 shows the geometry of this scenario.

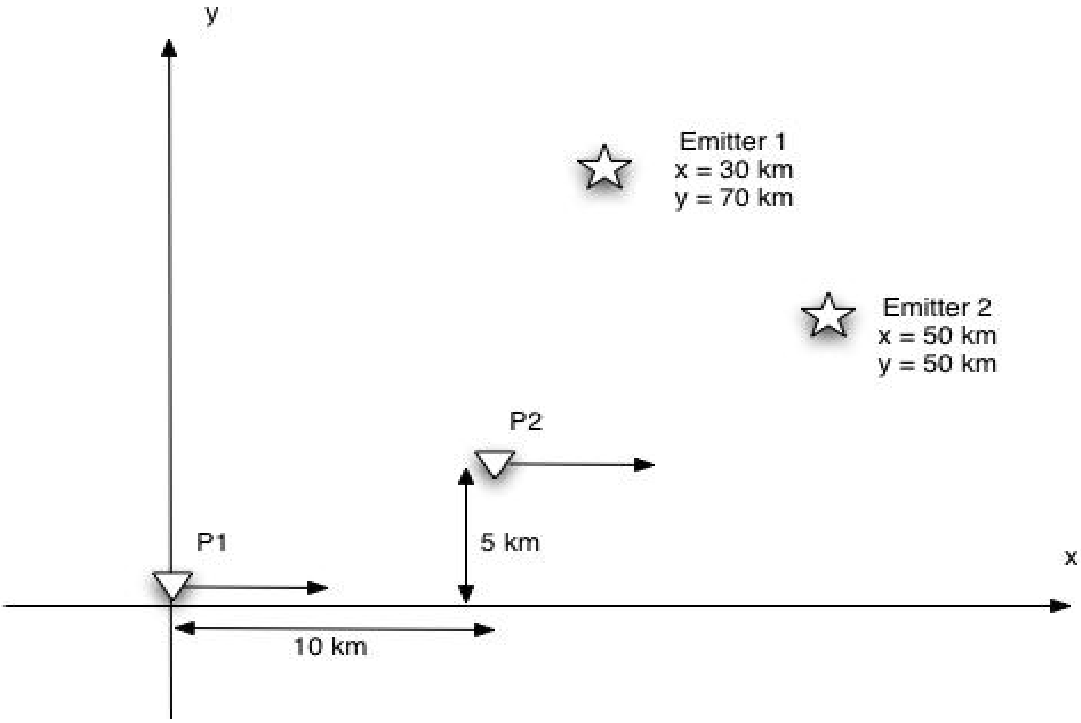


Figure 5-32: Collector Geometry for Scenario 3

The platforms were at an altitude of 7.5 km and are flying at 150 m/s. A snapshot was taken every one hundred seconds. The collectors moved 15 km between each snapshot along the x-axis. The Collector's snapshot setup for this series of snapshots follows:

- Carrier Frequency: 1000.025 MHz

Sampling Frequency:	100 kHz
Modulation Rate:	20 kbauds/sec
Modulation:	BPSK
Snap-duration:	65536 samples or 0.65536 seconds
Signal to Noise Ratio:	10 dB for each signal
Duration between snapshots:	100 seconds

These same settings were used to generate the signals from both emitters. Because the offset in the cross-track eliminates the left-right ambiguity only the positive portion of the area was mapped.

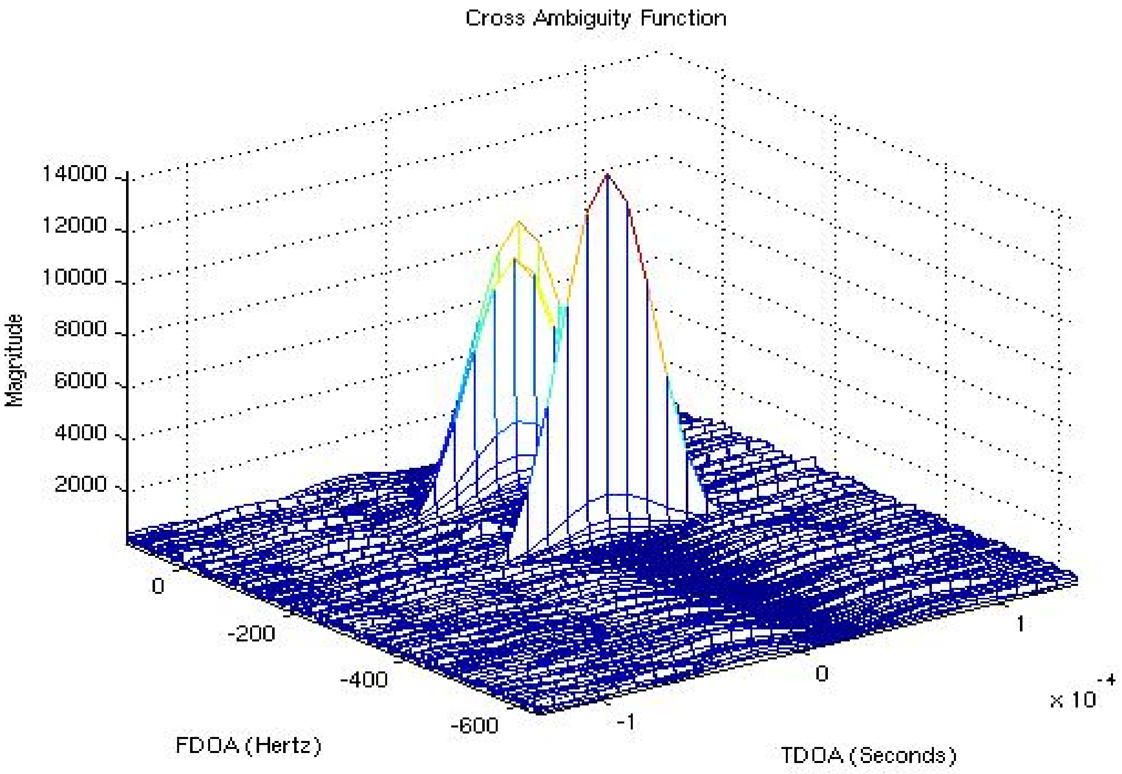


Figure 5-33: CAF of the First Snapshot

The CAF plane for the first snapshot shows not two peaks as expected but four. In Figure 5-33 it appears to have only two peaks but upon closer inspection in Figure 5-34 four FDOA peaks were noted. This is due to the seed generating the co-channel signal information is the same as the one that generated the original signal. This is a harder problem of two emitters that are not only co-channel but are also sending the same information. This means that the two emitters will be correlated at some time and frequency offset producing multiple peaks in the CAF surface.

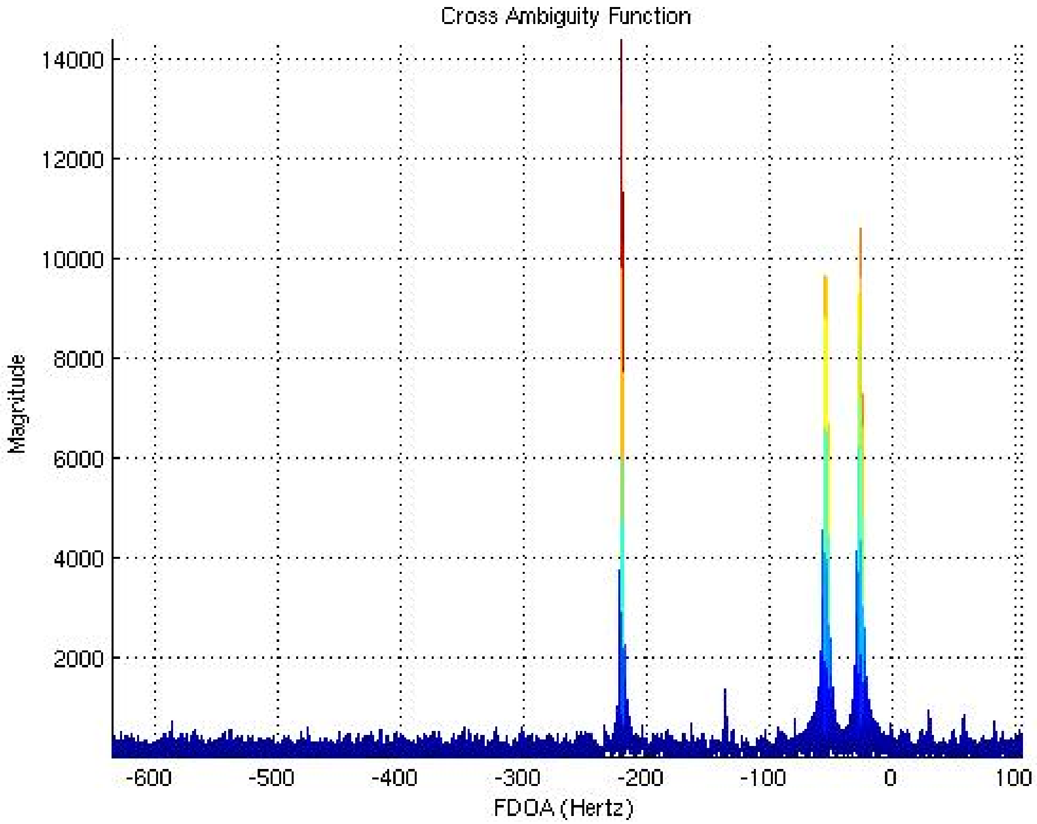


Figure 5-34: FDOA from the First Sanpshot

This produced three FDOA curves in the CAF-Map as illustrated in Figure 5-35. As the scenario continued these curves become clearer as the FDOAs began to separate.

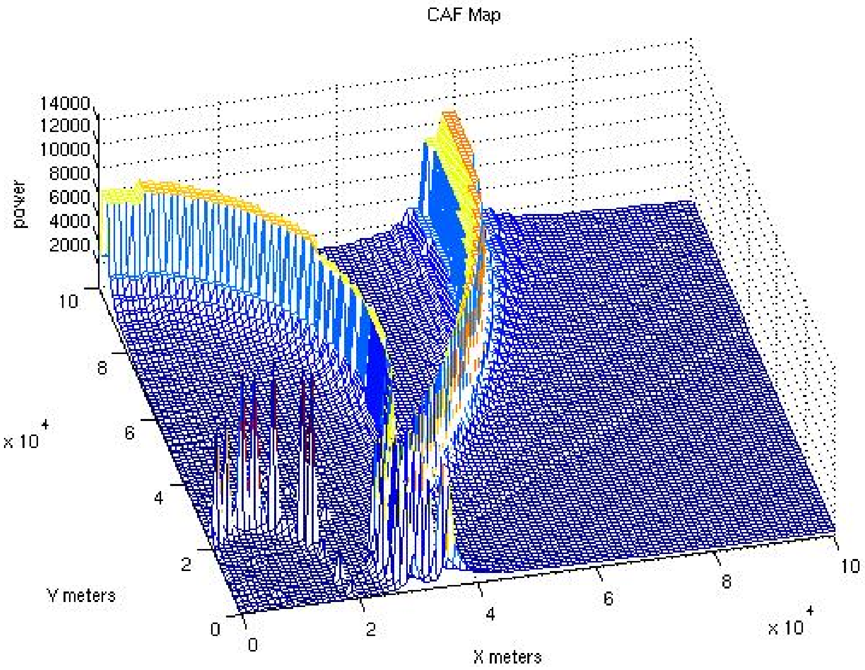


Figure 5-35: CAF-Map from collection pair at P1 = [5e3,0], P2 = [15e3,5e3] meters

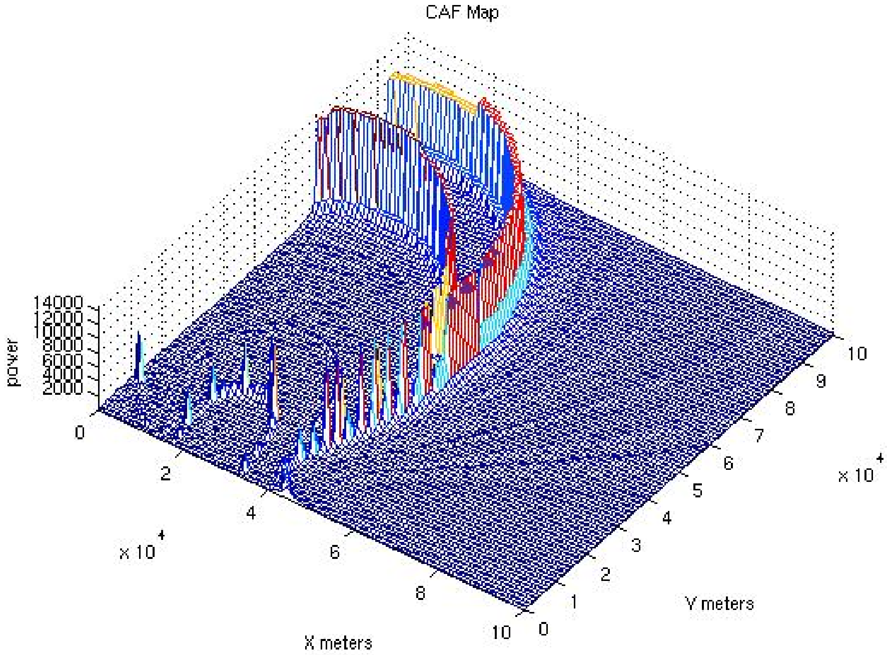


Figure 5-36: CAF-Map from collection pair at P1 = [20e3,0], P2 = [30e3,5e3] meters

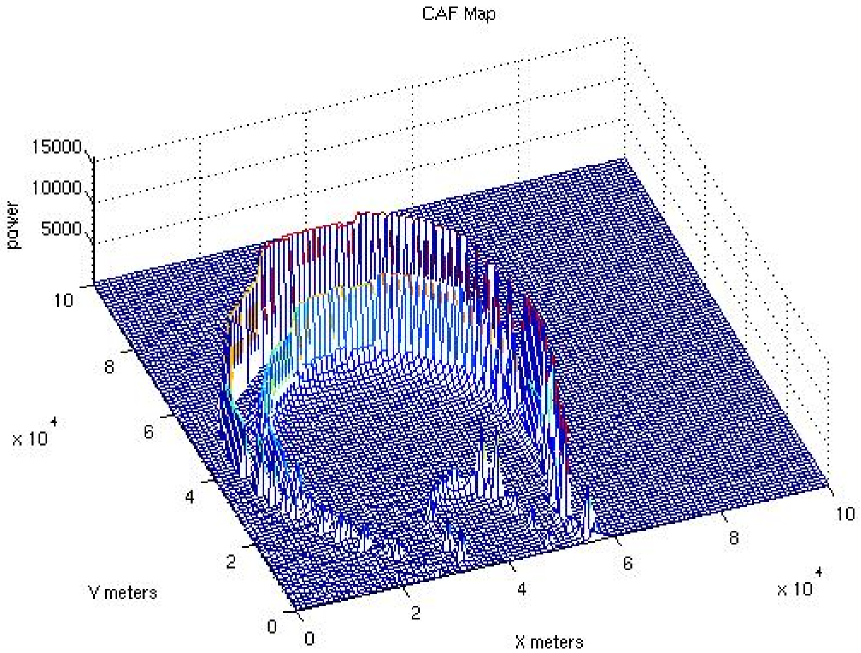


Figure 5-37: CAF-Map from collection pair at P1 = [35e3,0], P2 = [45e3,5e3] meters

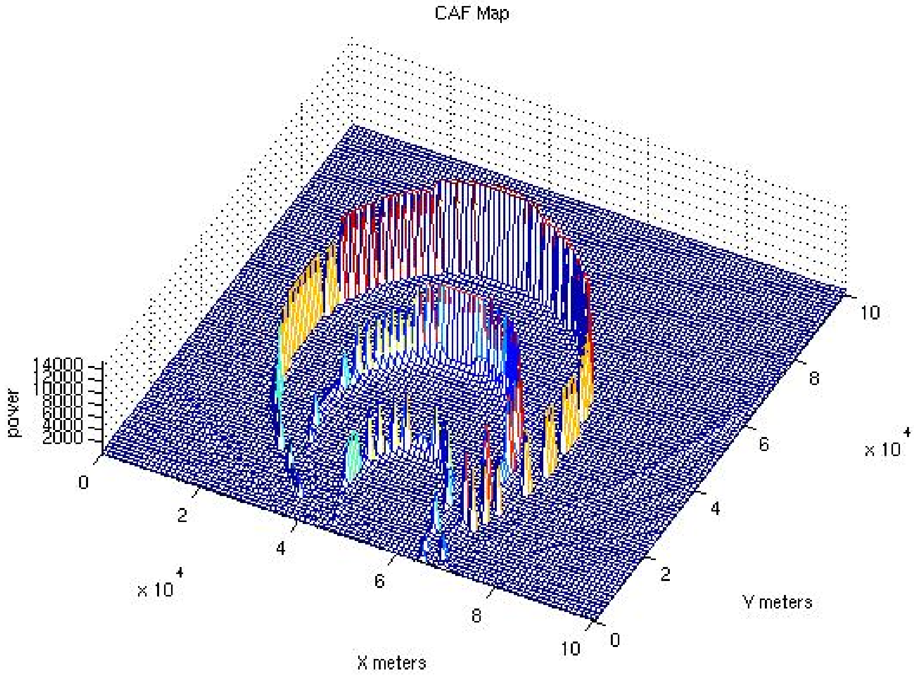


Figure 5-38: CAF-Map from collection pair at P1 = [50e3,0], P2 = [60e3,5e3] meters

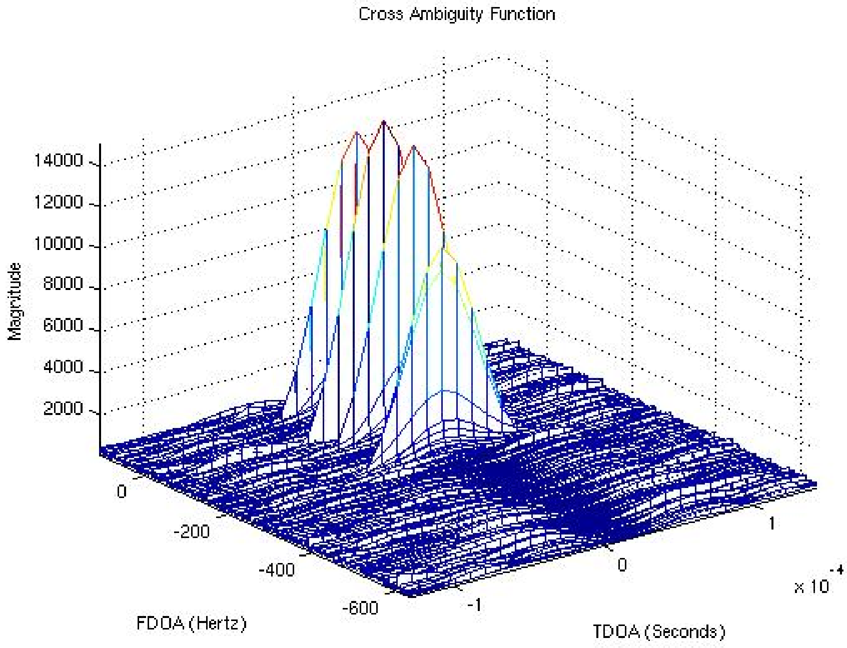


Figure 5-39: CAF Plane from Map shown in Figure 5-38

Note in Figure 5-39 that the snapshot while the collection pair was at P1 = [50e3,0] and P2 = [60e3,5e3] meters clearly shows the four peaks and Figure 5-40 shows the FDOAs.

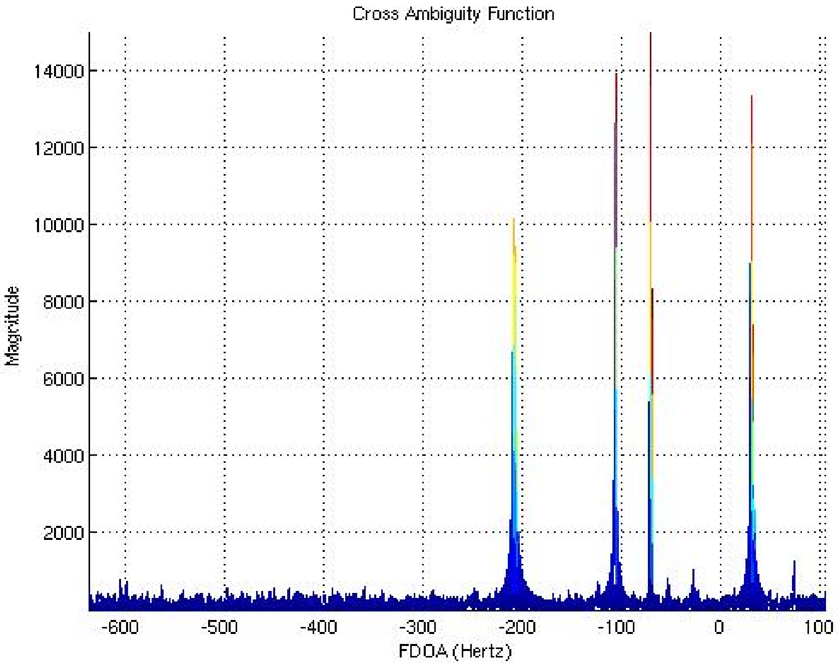


Figure 5-40: FDOAs of collection pair at P1 = [50e3,0], P2 = [60e3,5e3] meters

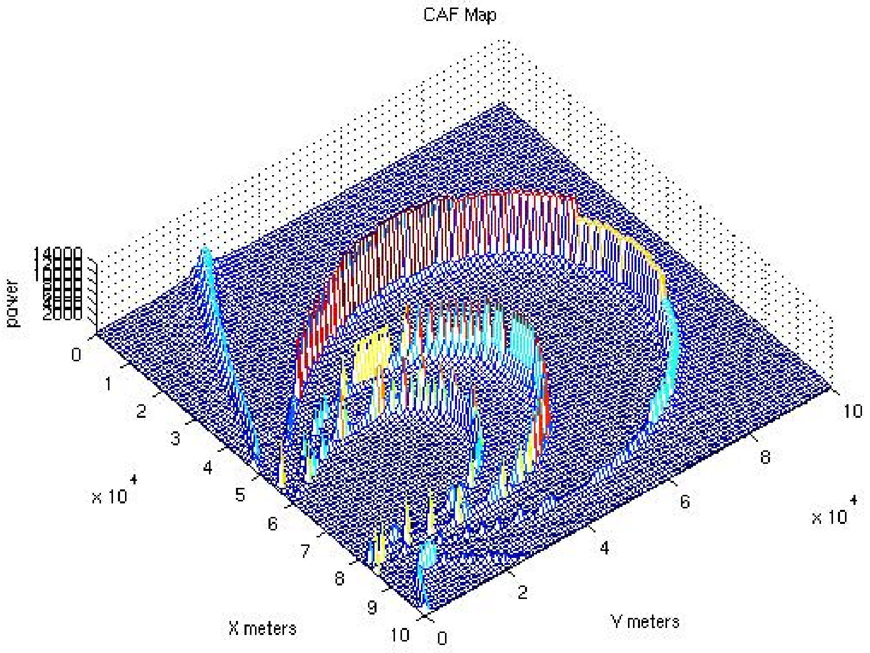


Figure 5-41: CAF-Map from collection pair at P1 = [65e3,0], P2 = [75e3,5e3] meters

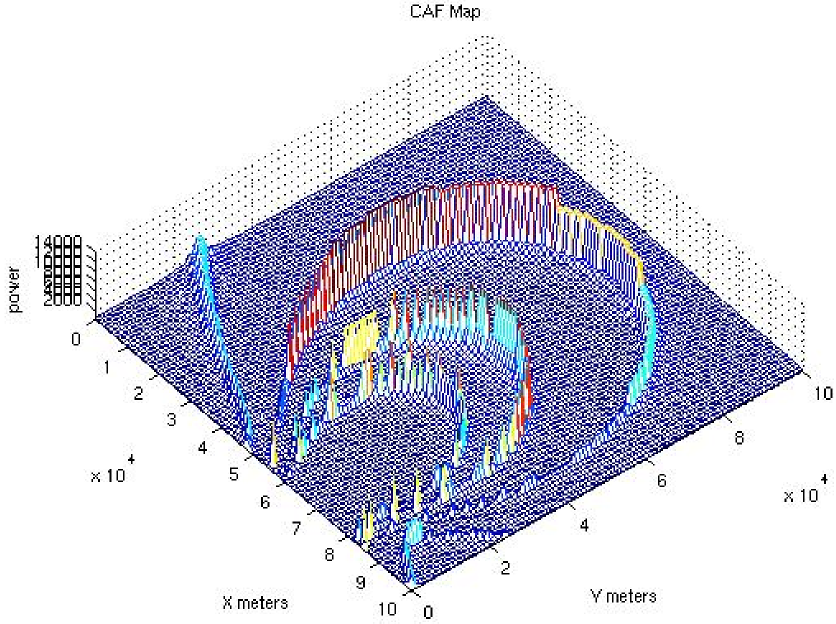


Figure 5-42: CAF-Map from collection pair at P1 = [80e3,0], P2 = [90e3,5e3] meters

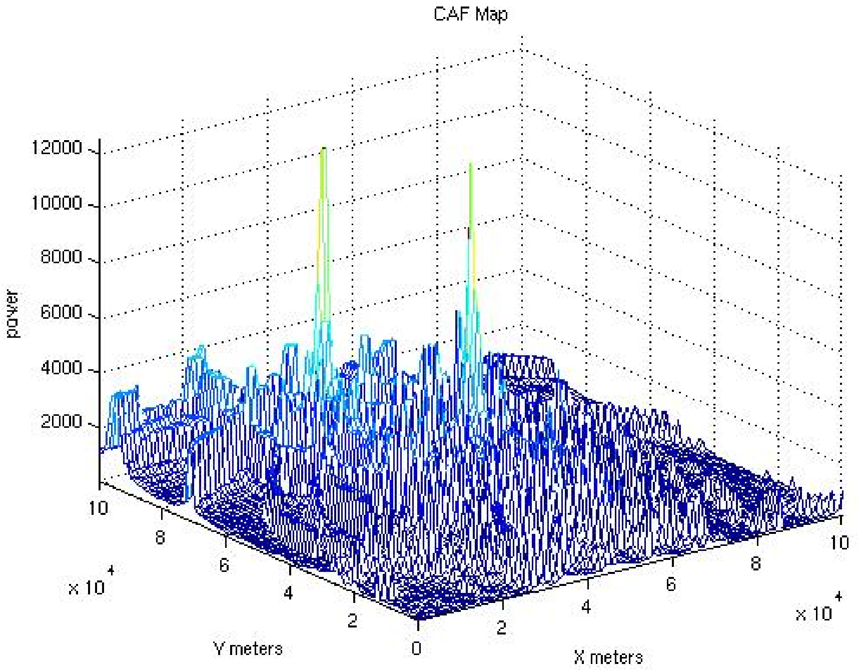


Figure 5-43: CAF-Map of the combined Maps

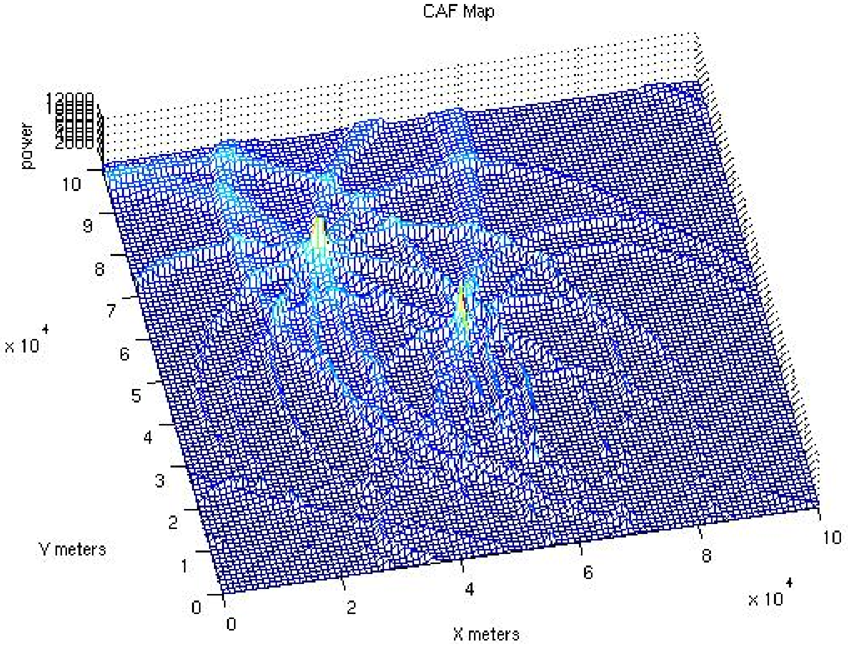


Figure 5-44: CAF-Map of the combined Maps

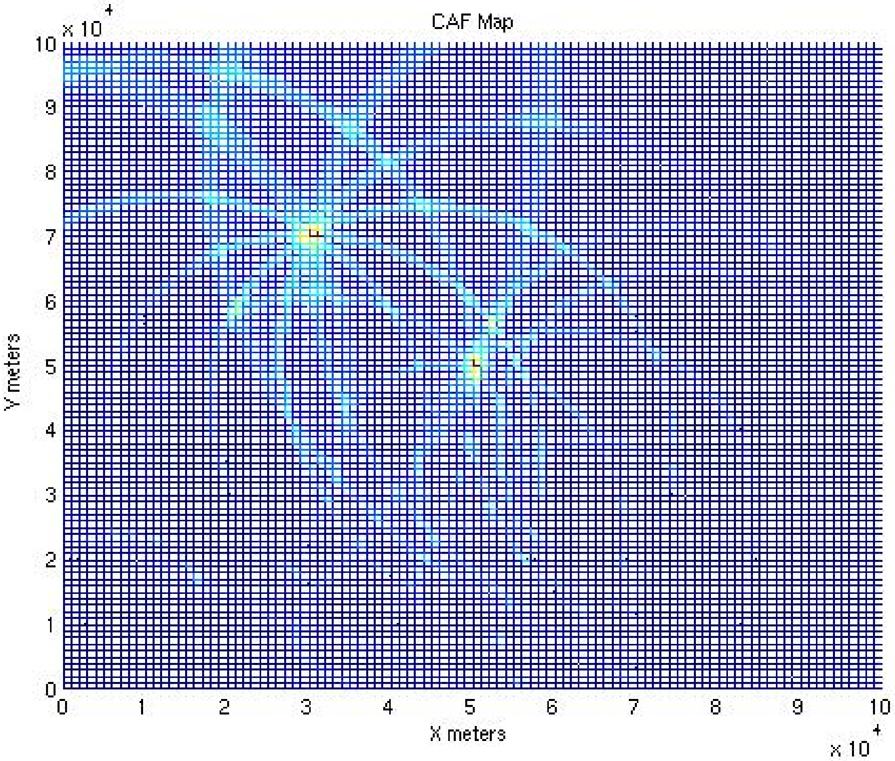


Figure 5-45: X-Y CAF-Map of the combined Maps

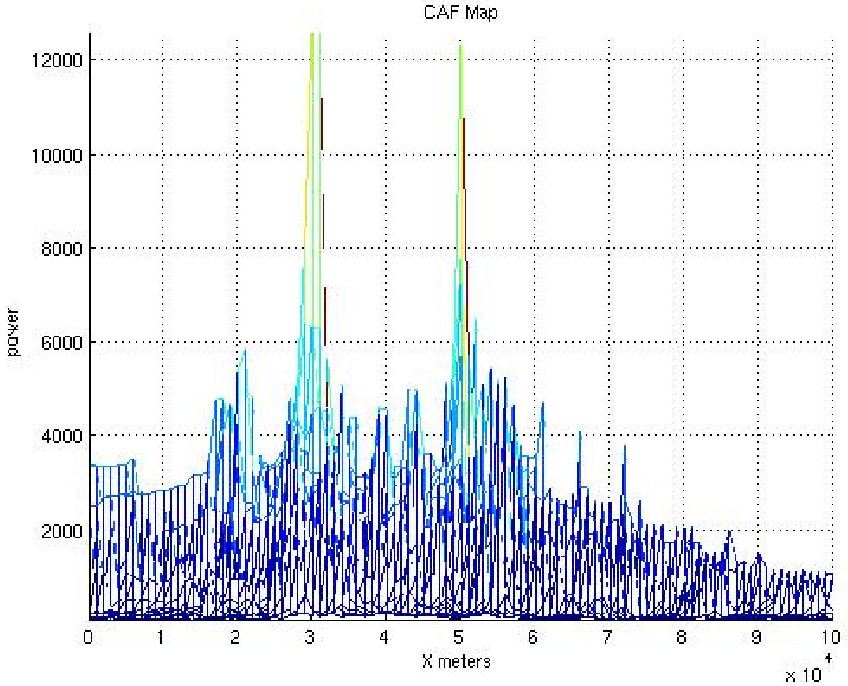


Figure 5-46: X-Z CAF-Map of the combined Maps

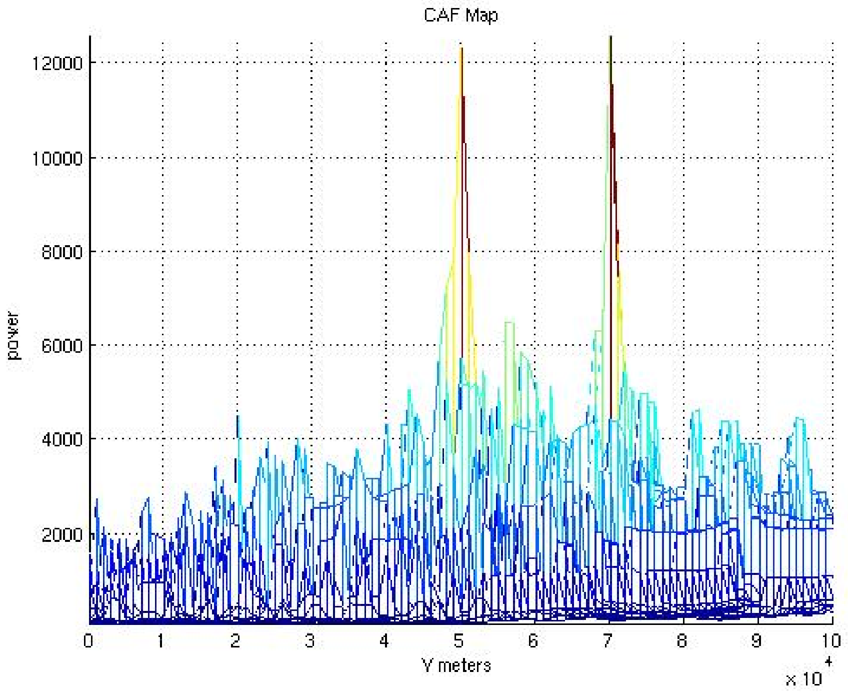


Figure 5-47: Y-Z CAF-Map of the combined Maps

As shown in Figures 5-43 through 5-47, the CAF-Map performed very well. Even with the CAF-Map resolution setting at 1000 meters and the interpolation turned off, the CAF-Map routine correctly located both emitters and had a miss distance of zero meters for each. Considering the emitters were correlated and co-channel, these results were remarkable.

D. SCENARIO #4

This scenario was developed to demonstrate the additional co-channel capability of the CAF-Map method. This scenario is an expansion of Scenario 3; it uses the same collection geometry and adds an additional target to the two that were used in Scenario 3. The first emitter was located at x = 30 km, y = 70 km, the second emitter was located at x = 50 km, y = 50 km, and the third emitter was located at x = 60 km, y = 70 km in the scene. As in Scenario 3, to combat the left-right ambiguity problem the collection platforms were separated along the cross-track by 5 km as well as along the ground-track by 10 km. The straight-line separation was 11,180 meters. The CAF-Map resolution for this scenario was set to 1000 meters and the CAF interpolation was turned off. Figure 5-

48 shows the geometry of this scenario. All three emitters are identical in modulation, SNR, modulation rate, and information being transmitted.

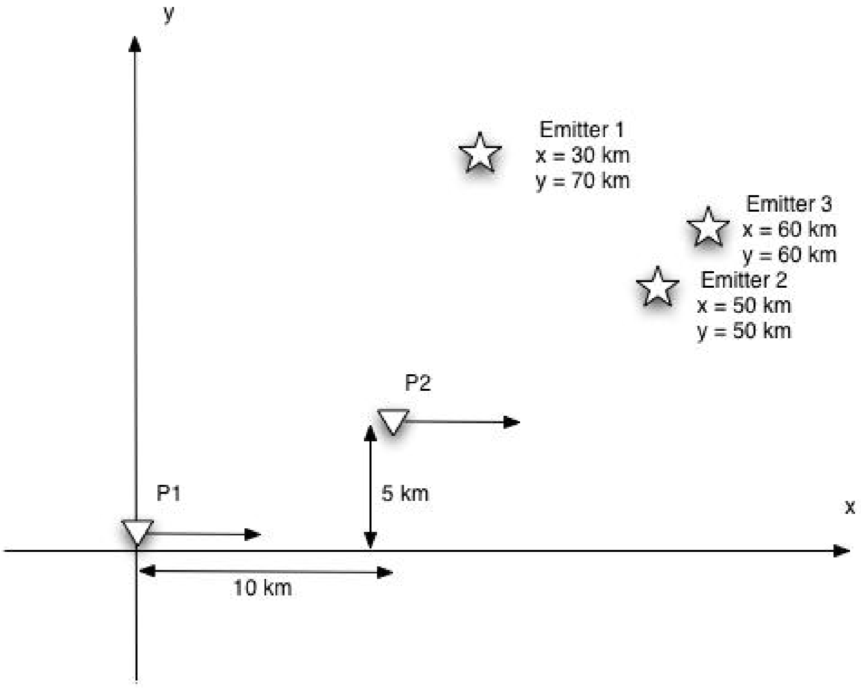


Figure 5-48: Collector Geometry for Scenario 4

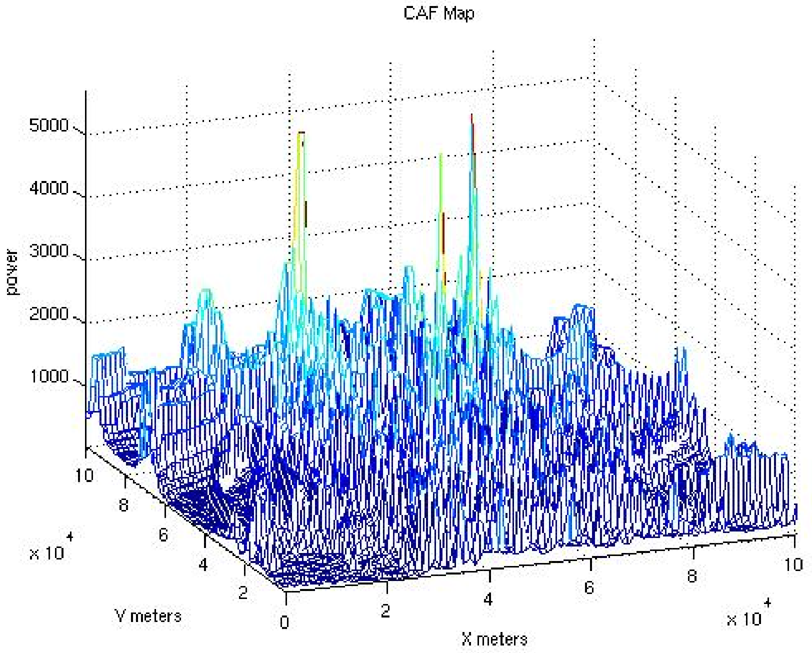


Figure 5-49: CAF-Map of the combined Maps

Figures 5-49, 5-50, 5-51, and 5-52 show the resulting CAF-Map images of the combined images

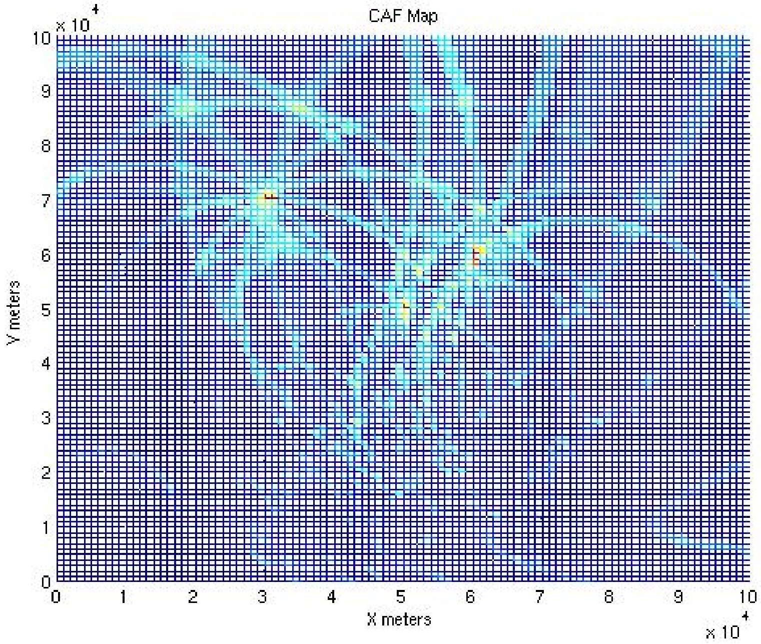


Figure 5-50: X-Y CAF-Map of the combined Maps

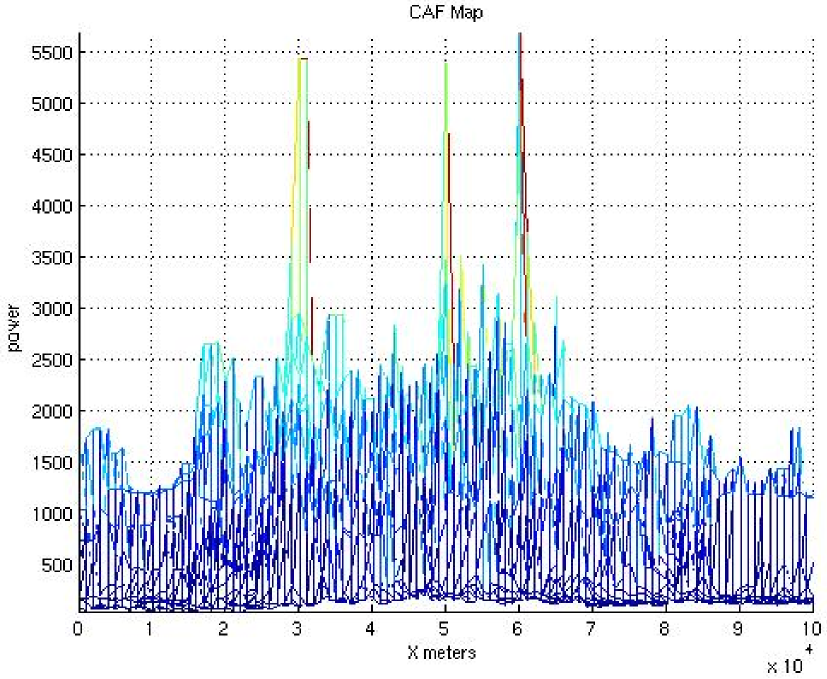


Figure 5-51: X-Z CAF-Map of the combined Maps

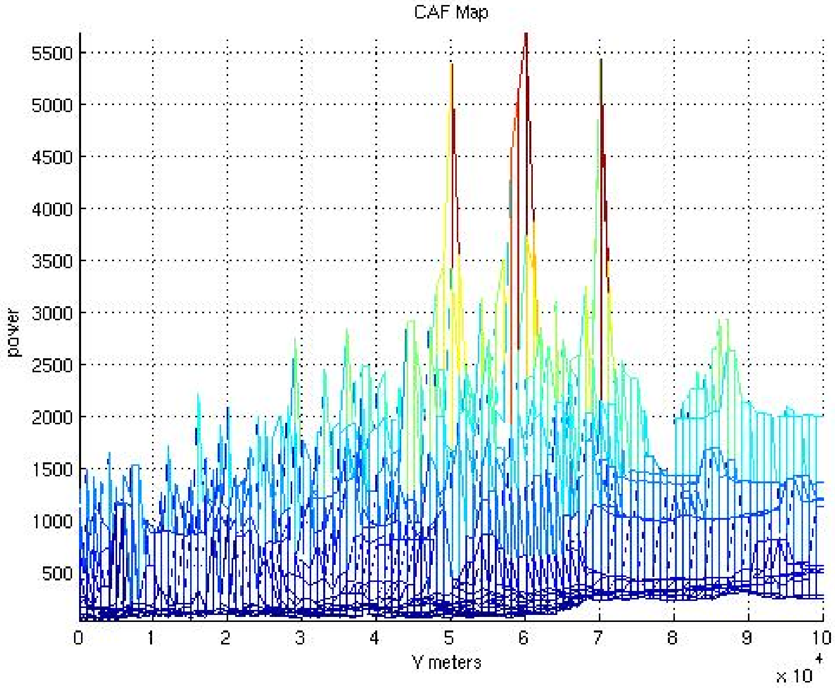


Figure 5-52: Y-Z CAF-Map of the combined Maps

The individual maps are not shown for this scenario because they are very similar to Scenario 3. The miss distances for all three targets were zero meters. However, at 1000-meter resolution, the peak of target 1 and target 3 are elongated in one direction to cover 2 grid points. Also note the added noise to the "floor" of the map. This is due to the multiple FDOA lines in the map that do not add to a point.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

A. SUMMARY OF FINDINGS

The goal of this thesis was to implement the CAF-Map method in MATLAB® and demonstrate this method's ability to geolocate emitters as an alternative to the traditional TDOA and FDOA geolocation methods. The advantage of this method lies in its ability to geolocate several co-channel emitters where traditional methods would have simply chosen the largest peak in the CAF surface or geolocated not only the true emitter location, but, several false locations by geolocating all the peaks in the CAF plane.

The main finding in this thesis is that the CAF-Map method can successfully geolocate up to three co-channel emitters. This thesis also reinforced the importance of the collector geometry asymmetry and its role in eliminating the left-right ambiguity effect.

B. FUTURE WORK

There are several ways that future work could build upon this thesis. The most obvious area would be to use a Digital Terrain Elevation Data (DTED) to generate the TDOA and FDOA look-up tables, providing a three-dimensional ground surface to improve the mapping of the TDOA and FDOA value to the ground. Additional resolution could be obtained by using a high resolution CAF function.

An active area of research is to apply super resolution methods to the CAF plane to increase the resolution. These super resolution methods could also improve the resolution of the CAF-Map image.

Additionally, work is still required to derive the geolocation error equations for this method.

An alternative method to the CAF-Map processes could be envisioned where the variables τ and f in the CAF equation are substituted with functions in latitude and longitude for a region that will allow a continuous time-like approach eliminating the requirement of breaking the collection into snapshots. This would allow a true coherent process that has not been possible while working with snapshot signals.

Additionally, the CAF-Map approach could also be applied to other geolocation methods such as phase interferometery, FDOA only, TDOA only, or Doppler geolocation techniques.

APPENDIX

This Appendix contains all the MATLAB$^{\circledR}$ functions and scripts used in this thesis. MATLAB$^{\circledR}$ Version 7, R14 was used in this thesis.

A. “CAF_MAP.M”

```
function [map,PtempX,PtempY]=caf_map(S1,S2,Fo,Fs,dm,Pe1,Pe2,Pc1,Vc1,Pc2,Vc2);
%*********************************************************************
%  [map,PtempX,PtempY]=caf_map(S1,S2,Fo,Fs,dm,Pe1,Pe2,Pc1,Vc1,Pc2,Vc2)
%  This function will calculate a CAF surface based upon input signals
%  S1 & S2 and map the caf to the 2 dimensional plane given by Pe1 and Pe2
%  Inputs:
%  S1          Signal from collector 1
%  S2          Signal from collector 2
%  Fo          Carrier Frequency of signal
%  Fs          Sampling Rate
%  dm          resolution in meters
%             Note: this depends on sample rate
%             and duration of the snapshot
%  Pe1         Start of grid for Emitter's Position [X,Y] in meters
%  Pe2         End of grid for Emitter’s Position {X,Y]
%  Pc1         Position of collector 1 and beginning of snapshot [X,Y,Z]
%  Vc1         Velocity Vector of collector 1 [x,y,z]
%  Pc2         Position of collector 2 and beginning of snapshot [X,Y,Z]
%  Vc2         Velocity Vector of collector 2 [x,y,z]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Written by:  Glenn Hartwell
% Last modified:  14 Dec. 2004
% ********************************************************************


%   First calculate tdoa and fdoa grids
[tdoa_grid, fdoa_grid, PtempX,PtempY] =
tdoa_fdoa_grid3D(Pc1,Vc1,Pc2,Vc2,Pe1,Pe2,Fo,dm);

%   Calculate min & max tdoa & fdoa

Tau_Lo = round(min(min(tdoa_grid))*Fs)-10;
Tau_Hi = round(max(max(tdoa_grid))*Fs)+10;
Freq_Lo = (min(min(fdoa_grid))/Fs)-.001;
Freq_Hi = (max(max(fdoa_grid))/Fs)+.001;
```

```
%    Calculate CAF... Using a modified version of LCDR Joe J. Johnson's caf_peak
function
[TDOA, FDOA, MaxAmb, Amb, TauValues,FreqValues] = ...
   CAF_peak(S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi, Fs, 0); %use 10 for intpr 0 if
no interp needed

%   Map CAF to X,Y Coordinates
[map,PtempX,PtempY] =
map_tdoa_fdoa(tdoa_grid,fdoa_grid,Amb,dm,Fs,TauValues,FreqValues,Pe1,Pe2);
```

B. “TDOA_FDOA_GRID3D.M”

```
function [tdoa_grid, fdoa_grid, indexX,indexY] =
tdoa_fdoa_grid3D(Pc1,Vc1,Pc2,Vc2,Pe1,Pe2,f0,dm);
% ***********************************************************************
% [tdoa_grid,fdoa_grid,indexX,indexY]=
tdoa_fdoa_grid3D(Pc1,Vc1,Pc2,Vc2,Pe1,Pe2,f0,dm)
%  Outputs:
%  tdoa_grid    Time Difference of arrival matrix for each grid
%  fdoa_grid    Frequency Difference of arrival matrix for each grid
%  indexX       X dimension index
%  indexY       Y Dimension index
%  Inputs:
%  Pc1      Collector one's Position [X,Y,Z] in meters
%  Vc1      Collector one's Velocity Vector [Vx,Vy,Vz] in meters/sec
%  Pc2      Collector two's Position {X,Y,Z] in meters
%  Vc2      Collector two's Velocity Vector [Vx,Vy,Vz] in meters/sec
%  Pe1      Start of grid for Emitter's Position [X,Y] in meters
%  Pe2      End of grid for Emitter’s Position {X,Y]
%  f0       Emitter's frequency in Hz
%  dm       resolution in meters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          this function generates tdoa and fdoa pairs based upon
%           Emitter frequency and Cartesian emitter-collector geometries.
%           The function returns two matrices:
%           tdoa & fdoa.
%
% Written by:  Glenn Hartwell
% Last modified:  21 Mar. 2004
% ***********************************************************************

c = 2.997925e8; % Speed of light in m/s
Ve = 0; %assume grid point has zero velocity
% Builds the position vectors for the Emitter's Position
% Note this assumes a flat earth and the emitter is a 0 alt
indexX = Pe1(1):dm:Pe2(1);  % X grid points
indexY = Pe1(2):dm:Pe2(2);  % Y grid points

Nx = length(indexX);
Ny = length(indexY);

for i = 1:Nx
   for j = 1:Ny
      % The next two lines calculate the Doppler shifts between the grid points
      % and Collector 1 & Collector 2, respectively for each point on the emitter grid
```

```
        gridP = [indexX(i),indexY(j),0];   % adds the 3rd dimensions at 0 meters in altitude

        doppler1(i,j) = f0/c * dot(Ve-Vc1, gridP-Pc1) / norm(gridP - Pc1);
        doppler2(i,j) = f0/c * dot(Ve-Vc2, gridP-Pc2) / norm(gridP - Pc2);

        % Calculates the FDOA

        fdoa_grid(i,j) = doppler1(i,j) - doppler2(i,j);


        % Calculates the TDOA

        tdoa_grid(i,j) = -(norm(gridP - Pc2) - norm(gridP - Pc1)) / c;
    end

end
```

C. “CAF_PEAK.M”

```
function [TDOA, FDOA, MaxAmb, Amb, TauValues,FreqValues] = ...
           CAF_peak(S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi, fs, intp);

% *********************************************************************
% CAF_peak(S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi, Fs, intp) takes as input:
%     two signals (S1, S2) that are row or column vectors; a range of
%     time delays (in samples) to search (Tau_Lo, Tau_Hi must be
%     integers between -N & +N); a range of digital frequencies (in
%     fractions of sampling frequency) to search (Freq_Lo, Freq_Hi must
%     be between -1/2 and 1/2, or -(N/2)/N and (N/2)/N, where N is the
%     length of the longer of the two signal vectors); and the sampling
%     frequency, fs.
%       [TDOA, FDOA, MaxAmb, Amb] = ...
%                   CAF_peak(S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi, fs);
%     The function computes the Cross Ambiguity Function of the two
%     signals. Four plots are produced which represent four different
%     views of the Cross Ambiguity Function magnitude versus the input
%     Tau and Frequency Offset ranges.
%
%     The function returns the scalars TDOA, FDOA, and MaxAmb, where
%     TDOA & FDOA are the values of Time Delay and Frequency Offset
%     that cause the Cross Ambiguity Function to peak at a magnitude
%     of MaxAmb.  Amb is the matrix of values representing the CAF
%     surface.

% Written by:  LCDR Joe J. Johnson, USN
% Modified by Glenn Hartwell
% 14 Dec. 2004
% *********************************************************************


% Ensures that the user enters all SIX required arguments.
if (nargin < 6)
  error...
  ('6 arguments required: S1, S2, Tau_Lo, Tau_Hi, Freq_Lo, Freq_Hi');
end

% Ensures that both S1 & S2 are row- or column-wise vectors.
if ((size(S1,1)~=1)&(size(S1,2)~=1)) | ((size(S2,1)~=1)&...
                                       (size(S2,2)~=1))
  error('S1 and S2 must be row or column vectors.');
end

N1 = length(S1);
```

```
N2 = length(S2);
S1 = reshape(S1,N1,1);    % S1 & S2 are reshaped into column-wise
S2 = reshape(S2,N2,1);    % vectors since MATLAB is more efficient
                     % when manipulating columns.

S1 = [S1;zeros(N2-N1,1)];   % Ensure that S1 & S2 are the same size,
S2 = [S2;zeros(N1-N2,1)];   % padding the smaller one w/ 0s as needed.


% This WHILE loop simply ensures that the length of S1 & S2 is a power
% of two.  If not, the vectors are padded with 0s until their length
% is a power of two.  This is not required, but it takes advantage of
% the fact that MATLAB's FFT computation is significantly faster for
% lengths which are powers of two!
while log(length(S1))/log(2) ~= round(log(length(S1))/log(2))
  S1(length(S1)+1) = 0;
  S2(length(S2)+1) = 0;
end

N = length(S1);

% Ensures that the Tau values entered are in the valid range.
if abs(Tau_Lo)>N | abs(Tau_Hi)>N
  error('Tau_Lo and Tau_Hi must be in the range -N to +N.');
end

% Ensures that Tau values entered by the user are integers.
if (Tau_Lo ~= round(Tau_Lo)) | (Tau_Hi ~= round(Tau_Hi))
  error('Tau_Lo and Tau_Hi must be integers.')
end

% Ensures that the Frequency values entered are in the valid range.
if abs(Freq_Lo)>1/2 | abs(Freq_Hi)>1/2
  error('Freq_Lo and Freq_Hi must be in the range -.5 to +.5');
end

% Ensures that the lower bounds are less than the upper bounds.
if (Tau_Lo > Tau_Hi) | (Freq_Lo > Freq_Hi)
  error('Lower bounds must be less than upper bounds.')
end

% Freq values converted into integers for processing.
Freq_Lo = round(Freq_Lo*N);
Freq_Hi = round(Freq_Hi*N);
```

```
% Creates vectors for the Tau & Freq values entered by the user. Used
% for plotting...
TauValues = [Tau_Lo:Tau_Hi];
FreqValues = [Freq_Lo:Freq_Hi]/N;

% The IF statement calculates the indices required to isolate the
% user-defined frequencies from the FFT calculations below.
if Freq_Lo < 0 & Freq_Hi < 0
   Neg_Freq = (N+Freq_Lo+1:N+Freq_Hi+1);
   Pos_Freq = [];
elseif Freq_Lo < 0 & Freq_Hi >= 0
   Neg_Freq = (N+Freq_Lo+1:N);
   Pos_Freq = (1:Freq_Hi+1);
else
   Neg_Freq = [];
   Pos_Freq = (Freq_Lo+1:Freq_Hi+1);
end


% This FOR loop actually calculates the Cross Ambiguity Function for
% the given range of Taus and Frequencies.  Note that an FFT is
% performed for each Tau value and then the frequencies of interest
% are isolated using the Neg_Freq and Pos_Freq vectors obtained above.
% For each value of Tau, the vector S2 is shifted Tau samples using a
% call to the separate function "SHIFTUD".  Samples shifted out are
% deleted and zeros fill in on the opposite end.

% Initializing Amb with 0s makes computations much faster.
Amb=zeros(length(Neg_Freq)+length(Pos_Freq),length(TauValues));
for t = 1:length(TauValues)
   temp = fft((S1).*conj(shiftud(S2,TauValues(t),0)));
   Amb(:,t) = [temp(Neg_Freq);temp(Pos_Freq)];
end


 if intp~=0
    interp = 1/intp;
    [xa,ya]=meshgrid(Tau_Lo:1:Tau_Hi,Freq_Lo:1:Freq_Hi);
    [xp,yp]=meshgrid(Tau_Lo:interp:Tau_Hi,Freq_Lo:interp:Freq_Hi);
    Zp = interp2(xa,ya,Amb,xp,yp,'cubic');
    TauValues = [Tau_Lo:interp:Tau_Hi];
    FreqValues = [Freq_Lo:interp:Freq_Hi]/N;
    figure
    mesh(TauValues/fs,FreqValues*fs,abs(Zp));
    xlabel('TDOA (Seconds)');ylabel('FDOA (Hertz)');
    zlabel('Magnitude');
    title('Cross Ambiguity Function');
    axis tight
    Amb = Zp;
else
    figure      % This one is the 3-D view
    mesh(TauValues/fs,FreqValues*fs,abs(Amb));
    xlabel('TDOA (Seconds)');ylabel('FDOA (Hertz)');
    zlabel('Magnitude');
    title('Cross Ambiguity Function');
    axis tight

end
% Only interested in the Magnitude of the Cross Ambiguity Function.
abs_Amb = abs(Amb);


% Finds the indices of the peak value.
[DFO, DTO] = find(Amb==max(max(abs_Amb)));

TDOA = TauValues(DTO);  % Finds the actual value of the TDOA.
FDOA = FreqValues(DFO); % Finds the actual value of the FDOA.
MaxAmb = max(max(abs_Amb)); % Finds the actual Magnitude of the peak.
```

D. “MAP_TDOA_FDOA.M”

```
function [map,PtempX,PtempY] =
map_tdoa_fdoa(tdoa_grid,fdoa_grid,G,dm,Fs,TauValues,FreqValues,Pe1,Pe2);
% ********************************************************************
% [map]= map_tdoa_fdoa(tdoa_grid,fdoa_grid,G,dm,blocksize,fftsize,Pe1,pe2)
%  Outputs:
%  map          map of the tdoa and fdoa mapped to the ground(x,y)
%  Inputs:
%  tdoa_grid   tdoas of each x,y
%  fdoa_grid   fdoa of each x,y
%  G            Caf in tdoa and fdoa plane
%  dm           resolution in meters for fdao and tdoa grid
%  Fs           Sample freq
%  Pe1          Start of grid for Emitter's Position [X,Y] in meters
%  Pe2          End of grid for Emitter’s Position {X,Y]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%         this function generates a caf mapped to the x,y plane using
%         the outputs of tdoa_fdoa_grid3D and CAF functions
%         This function includes the function findnearest by By Tom Benson (2002)
%         of University College London
%
% Written by:  Glenn Hartwell
% Last modified:  14 Dec. 2004
% ********************************************************************

[m,n] = size(tdoa_grid);
%fdoa and tdoa grid are the same size
[u,v] = size(G);

for x = 1:m
   for y = 1:n
      t = tdoa_grid(x,y);
      f = fdoa_grid(x,y);
      j = findnearest(TauValues,(t*Fs),0);
      i = findnearest(FreqValues,(f/Fs),0);
      map(x,y)=G(i,j);
   end
end

% Mesh result
PtempX = Pe1(1):dm:Pe2(1);  % X grid points
PtempY = Pe1(2):dm:Pe2(2);  % Y grid points
figure
f=mesh((abs(map')));
```

```
set(f,'XData',PtempX);
set(f,'YData',PtempY);
xlabel('X meters');
ylabel('Y meters');
zlabel('power');
axis tight
title('CAF Map');
```

E. “FINDNEAREST.M”

```
function [r,c,V] = findnearest(srchvalue,srcharray,bias)

% Usage:
% Find the nearest numerical value in an array to a search value
% All occurances are returned as array subscripts
%
% Output:
%
% For 2D matrix subscripts (r,c) use:
%
%       [r,c] = findnearest(srchvalue,srcharray,gt_or_lt)
%
%
% To also output the found value (V) use:
%
%       [r,c,V] = findnearest(srchvalue,srcharray,gt_or_lt)
%
%
% For single subscript (i) use:
%
%         i   = findnearest(srchvalue,srcharray,gt_or_lt)
%
%
% Inputs:
%
%    srchvalue = a numerical search value
%    srcharray = the array to be searched
%    bias      = 0 (default) for no bias
%                -1 to bias the output to lower values
%                 1 to bias the search to higher values
%                (in the latter cases if no values are found
%                 an empty array is ouput)
%
%
% By Tom Benson (2002)
% University College London
% t.benson@ucl.ac.uk

if nargin<2
    error('Need two inputs: Search value and search array')
elseif nargin<3
    bias = 0;
end
```

```
% find the differences
srcharray = srcharray-srchvalue;

if bias == -1   % only choose values <= to the search value

    srcharray(srcharray>0) =inf;

elseif bias == 1  % only choose values >= to the search value

    srcharray(srcharray<0) =inf;

end

% give the correct output
if nargout==1 | nargout==0

    if all(isinf(srcharray(:)))
        r = [];
    else
        r = find(abs(srcharray)==min(abs(srcharray(:))));
    end

elseif nargout>1
    if all(isinf(srcharray(:)))
        r = [];c=[];
    else
        [r,c] = find(abs(srcharray)==min(abs(srcharray(:))));
    end

    if nargout==3
        V = srcharray(r,c)+srchvalue;
    end
end
```

F. “SIG_GEN.M”

```
 function [Sa1, Sa2, S1, S2, Pcc1, Pcc2] = sig_gen;

% ***********************************************************************
% [Sa1, Sa2, S1, S2, Pc1, Pc2]] = sig_gen;
% SIG_GEN generates BPSK signal pairs based upon user-defined param-
%           eters and Cartesian emitter-collector geometries.  There are
%           no input arguments, since the function queries the user for
%           all required inputs.  The function returns four vectors:
%           Sa1, Sa2, S1 & S2.  These are the Analytic Signal represen-
%           tations of the two generated signals, and the Real represen-
%           tations of the two signals, respectively.
%
% Written by:  LCDR Joe J. Johnson, USN
% Last modified:  28 June 2005 By Glenn D. Hartwell
% Modified for 3D simulations and export collectors positions
% ***********************************************************************

clc;
disp(' ');
disp('All positions and velocites must be entered in vector format,');
disp('e.g., [X Y Z] or [X, Y, Z] (including the brackets).');
disp(' ');

Pc1(1,:) = input...
      ('Collector 1''s POSITION Vector at time 0 (in meters)?  ');
Vc1 = input('Collector 1''s VELOCITY Vector (in m/s)?  '); disp(' ');

Pc2(1,:) = input...
      ('Collector 2''s POSITION Vector at time 0 (in meters)?  ');
Vc2 = input('Collector 2''s VELOCITY Vector (in m/s)?  '); disp(' ');

Pe(1,:) = input...
          ('Emitter''s POSITION Vector at time 0 (in meters)?  ');
Ve = input('Emitter''s VELOCITY Vector (in m/s)?  '); disp(' ');

% f0 and fs are the same for BOTH collectors!
f0 = input('Carrier Frequency (in Hz)?  ');
fs = input('Sampling Frequency (in Hz)?  ');
Ts = 1/fs;      % Calculates Sample Period

Rsym = input('Symbol Rate (in symbols/s)?  '); disp(' ');
Tsym = 1/Rsym;                    % Calculates Symbol Period

N = input('How many samples?  '); disp(' ');
```

```
Es_No1 = input('Desired Es/No at Collector 1 (in dB)?  ');
Es_No1 = 10^(Es_No1/10);          % Converts from dB to ratio

Es_No2 = input('Desired Es/No at Collector 2 (in dB)?  ');
disp(' ');
Es_No2 = 10^(Es_No2/10);          % Converts from dB to ratio


Pc1 = [Pc1; zeros(N-1, 3)];     % Initializing all the matrices makes
Pe1 = zeros(N, 3);              % later computations much faster.
Pc2 = [Pc2; zeros(N-1, 3)];
Pe2 = zeros(N, 3);
t1 = zeros(1, N);
t2 = zeros(1, N);
S1 = zeros(1, N);
S2 = zeros(1, N);

A = 1;                 % Amplitude of Signal
c = 2.997925e8; % Speed of light in m/s
Ps = (A^2)/2;       % Power of Signal

sigma1 = sqrt(Ps*Tsym/Es_No1);  % Calculate Noise Amplification fac-
sigma2 = sqrt(Ps*Tsym/Es_No2);  % tors using Es/No = Ps*Tsym/sigma^2

Noise1 = sigma1.*randn(N, 1);      % Random Noise values for Signal 1
Noise2 = sigma2.*randn(N, 1);      % Random Noise values for Signal 2


% Builds the position vectors for the two collectors
for index = 2 : N
   Pc1(index,:) = Pc1(index - 1,:) + Ts*Vc1;
   Pc2(index,:) = Pc2(index - 1,:) + Ts*Vc2;
end


% While loop determines first elements of Pe1 and t1.  t1(1) is the
% time AT THE EMITTER that produces the 1st sample received at
% collector 1!  Pe1(1,:) is the position of the emitter when it
% produces the 1st sample received by collector 1.

temp = inf;        % Ensures while loop executes at least once
t1(1) = 0;
tempPe = Pe(1,:);
while abs(temp - t1(1)) > 1/f0
```

```
   temp = t1(1);
   t1(1) = -norm(Pc1(1,:) - tempPe) / c;
   tempPe = Pe(1,:) + t1(1)*Ve;
end
Pe1(1,:) = tempPe;


% While loop determines first elements of Pe2 and t2.  t2(1) is the
% time AT THE EMITTER that produces the 1st sample received at
% collector 2!  Pe2(1,:) is the position of the emitter when it
% produces the 1st sample received by collector 2.

temp = inf;         % Ensures while loop executes at least once
t2(1) = 0;
tempPe = Pe(1,:);
while abs(temp - t2(1)) > 1/f0
   temp = t2(1);
   t2(1) = -norm(Pc2(1,:) - tempPe) / c;
   tempPe = Pe(1,:) + t2(1)*Ve;
end
Pe2(1,:) = tempPe;

% Platform positions at middle of snapshot
Pcc1=(Pc1(N/2,:));
Pcc2=(Pc2(N/2,:));
% Determines the earliest time at the emitter for this pair of signals.
StartPoint = min(t1(1), t2(1));


% Next 2 lines determine offsets needed for signals 1 & 2 to enter the
% phase vector (P).  This simply ensures proper line up so that bit
% changes occur at the right times.
SymbolIndex1 = 1 + floor(abs(t1(1) - t2(1))/Tsym) * (t1(1)>t2(1));
SymbolIndex2 = 1 + floor(abs(t1(1) - t2(1))/Tsym) * (t2(1)>t1(1));


for index = 2 : N                 % Builds the Pe1 and t1 vectors
   temp = inf;
   t1(index) = 0;

   % 1st guess is that emitter will advance exactly Ts seconds.
   tempPe = Pe1(1,:) + (t1(index -1) + Ts)*Ve;

   % While loop iteratively determines actual time & position for
   % emitter, based on instantaneous geometry.
```

```
  while abs(temp - t1(index)) > 1/f0
    temp = t1(index);
    t1(index) = (index - 1)*Ts - norm(Pc1(index,:) - tempPe) / c;

    % Due to negative times, must multiply Ve by ELAPSED time!
   tempPe = Pe1(1,:) + abs(t1(1)-t1(index))*Ve;
  end
  Pe1(index,:) = tempPe;
end


for index = 2 : N               %Builds the Pe2 and t2 vectors
  temp = inf;
  t2(index) = 0;

  % 1st guess is that emitter will advance exactly Ts seconds.
  tempPe = Pe2(1,:) + (t2(index -1) + Ts)*Ve;

   % While loop iteratively determines actual time & position for
  % emitter, based on instantaneous geometry.
  while abs(temp - t2(index)) > 1/f0
    temp = t2(index);
   t2(index) = (index - 1)*Ts - norm(Pc2(index,:) - tempPe) / c;

    % Due to negative times, must multiply Ve by ELAPSED time!
    tempPe = Pe2(1,:) + abs(t2(1)-t2(index))*Ve;
  end
  Pe2(index,:) = tempPe;
end


% Could change this seed to whatever you want; or could have user
% define it as an input.  This just ensures, for simulation purposes
% that every time the program is run, the BPSK signals created will
% have the same random set of data bits.
rand('seed',5);

% Create enough random #'s to figure phase shift (data bits)
r = rand(N,1);
P = (r > 0.5)*0 + (r <= 0.5)*1;     % Since BPSK, random # determines
                                   % if phase is 0 or pi


% Building Xmitted Signal #1 vector...  These represent the pieces of
% the signal that were transmitted by the emitter to arrive at
% Collector 1 at its sample intervals.

S1(1) = A*cos(2*pi*f0*t1(1) + P(SymbolIndex1)*pi) + Noise1(1);

% The if statement inside the loop changes the data bit if the time
% has advanced into the next symbol period.
for index = 2 : N
  if t1(index) - StartPoint >= (SymbolIndex1) * Tsym
    SymbolIndex1 = SymbolIndex1 + 1;
  end
  S1(index) = A*cos(2*pi*f0*t1(index) + P(SymbolIndex1)*pi) + ...
           Noise1(index);
end

Sa1 = hilbert(S1);  % Calculates the ANALYTIC SIGNAL of S1.  To
                   % compute the COMPLEX ENVELOPE, multiply Sa1
                   % by .*exp(-j*2*pi*f0.*t1);


% Building Xmitted Signal #2 vector...  These represent the pieces of
% the signal that were transmitted by the emitter to arrive at
% Collector 2 at its sample intervals.

S2(1) = A*cos(2*pi*f0*t2(1) + P(SymbolIndex2)*pi) + Noise2(1);

% The if statement inside the loop changes the data bit if the time
% has advanced into the next symbol period.
for index = 2 : N
  if t2(index) - StartPoint >= (SymbolIndex2) * Tsym
    SymbolIndex2 = SymbolIndex2 + 1;
  end
  S2(index) = A*cos(2*pi*f0*t2(index) + P(SymbolIndex2)*pi) + ...
           Noise2(index);
end

Sa2 = hilbert(S2);  % Calculates the ANALYTIC SIGNAL of S2.  To
                   % compute the COMPLEX ENVELOPE, multiply Sa2
                   % by .*exp(-j*2*pi*f0.*t2);


% This function call simply calculates and displays the expected TDOAs
% and FDOAs at the Beginning and End of the collection time.

tdoa_fdoa(f0,Pe1(1,:),Pe1(N,:),Pe2(1,:),Pe2(N,:),Ve,Pc1(1,:),...
      Pc1(N,:),Vc1,Pc2(1,:),Pc2(N,:),Vc
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] A. Buczek, unpublished notes and private conversations at the Naval Research Laboratory Dec. 2002

[2] S. Stien, “Algorithms for Ambiguity Function Processing,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, No 3, pp. 588-599, Jun 1981

[3] R. Ulman & E. Geraniotis “Wideband TDOA/FDOA Processing Using Summation of Short-Time CAF’s,” *IEEE Transactions on Signal Processing*, vol. 47, No 12, pp 3193-3200 Dec. 1999

[4] J. Stoner, “Where is it,” LEO System Library NRL, Washington DC, Non-numbered technical report, 1991

[5] J. Stoner, “Dry Gulch Jake and the Goddess of the Desert (or KEEP THE APRIORI),” LEO System Library NRL, Washington DC, Stoner Memo: 129 Sep 2000

[6] H. Loomis, Jr., “Geolocation of Electromagnetic Emitters,” Technical Report No NPS-EC-00-003, Naval Postgraduate School, Monterey, CA, Nov 1999.

[7] M. Price, “Covariance and Information Matrices: A Primer,” LEO System Library NRL Washington DC, Memo, Jan 2004

[8] M. Price, “Least Squares Geolocation Data Combining – a Summary,” LEO System Library NRL Washington DC, Memo, 19 Aug. 2002

[9] G. Clark, “Simplified Determination of the Ellipse of Uncertainty,” *Navigation: Journal of The Institute of Navigation*, Vol. 21, No 4, pp 343-350, Winter 1974-75

[10] R. Daniels, “A note on combining Geolocation estimates,” LEO System Library NRL Washington DC, Non-numbered technical report, Aug. 1980

[11] A. Papoulis, *Probability, Random Variables, and Stochastic Processes. Third Edition.* New York: McGraw-Hill, 1991. pp 79, 253.

[12] J. Johnson, “Implementing the Cross Ambiguity Function and Generating Geometry- Specific Signals,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, Sep 2001

[13] G. Hartwell & S. Jordan, “MATLAB Implementation of the Complex Ambiguity Function,” paper presented at the SIGINT Systems I class project presentation, Naval Postgraduate School, Monterey, CA, 15 Sep 2001.

[14] Astronomical Society of the Pacific, *Synthesis Imaging in Radio Astronomy*, Volume 6, 1989

[15] Astronomical Society of the Pacific, *Synthesis Imaging in Radio Astronomy II*, Volume 180, 1999

[16] M. Schwäbisch, “A Fast and Efficient Technique for SAR Interogram Geocoding,” *Geoscience and Remote Sensing Symposium 1998, IGARSS'98,* IEEE 1998 International, vol. 2, pp 1100- 1102, 1998

[17] J. Wu, Y Huang and X. Dong, “Image Retrieval Algorithm of Two-Dimensional Synthetic Aperture Radiometer,” *Geoscience and Remote Sensing Symposium, 2001, IGARSS '01*, IEEE 2001 International, vol. 6, pp 3268-3270, 9-13 July 2001

[18] E. Sansosti, “A Simple and Exact Solution for the Interferometric and Stereo SAR Geolocation Problem,” IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no 8, pp 1625-1634, Aug. 2004

[19] M. Price, “Mathematics of Geolocation,” LEO System Library NRL Washington DC, Non-numbered technical report, Not dated

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Ruth H. Hooker Library
   Naval Research Laboratory
   Washington, DC

4. Chairman, ECE Department
   Naval Postgraduate School
   Monterey, CA